HARDWARE AND SOFTWARE SCIENTIFIC RESEARCH
AND TECHNOLOGICAL ENGINEERING INSTITUTE

HARDWARE DEPARTMENT

COBRA
HARDWARE DESCRIPTION

BRAŞOV

1988

EDITORIAL MANAGERS:

Dr. eng. **Dan Roman**

Dr. **Emil Munteanu**

COVER DESIGN:

designer **Liviu Derveşteanu**

The **COBRA** microcomputer was designed and built by the following team:
eng. **Vasile Prodan,** eng. **Sorin Finichiu,** eng. **Bernd Hansgeorg Wagner,**
math. **Marcel Arefia,** math. **Mircea Pop,** eng. **Mircea Ungur,**
arch. **Alexandru Antal.**

This material has been edited by: eng. **Vasile Prodan,**
eng. **Mircea Ungur.**

Coordinator: Dr. Eng. **Gheorghe Toacşe**

C O N T E N T S

1. FUNCTIONAL DESCRIPTION OF COBRA MICROCOMPUTER

1.1 BLOCK DIAGRAM

The **CoBra** microcomputer is built around the 8-bit microprocessor **Z80A**, on three circuit boards. The microcomputer block diagram is shown in **Fig.1**.

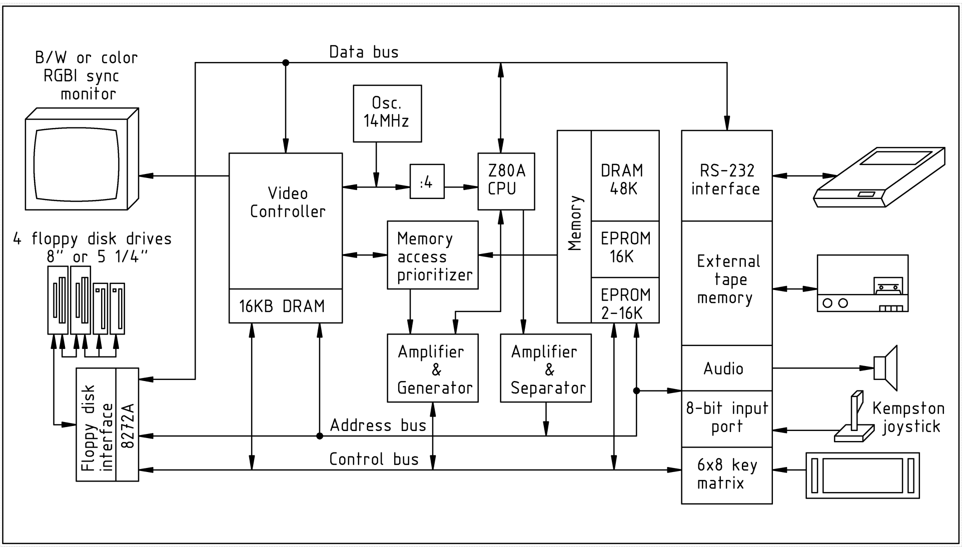


Fig. 1 – **CoBra** microcomputer block diagram

On the mainboard there are:

— the central processing unit block with the **Z80A** μP and address/control bus amplifiers/separators.

— the memory block consisting of a configuration/selection circuit, a 48 KB **DRAM** circuit and a 2-16 KB **EPROM** circuit.

— the video controller consisting of 16 KB **DRAM** video memory, memory access priority controller (memory access prioritizer), generators for video synchronization signals and composite color signals, video memory control signals generator, and the system clock generator.

— interface block, built around a parallel programmable interface **i8255**.

— **RS-232** interface for use with a printer or another computer.

— external memory (magnetic tape) interface.

— audio interface.

— keyboard interface (6x8 key matrix).

— 8 bit input port for general use or used as a **Kempston**-compatible joystick interface.

The conventional keyboard is built on a separate board, as an extended 6x8 key matrix plus **ZX SPECTRUM+** compatible compound keys.

The floppy disk interface is built on a third board around the **i8272A** Floppy Disk Controller circuit assisted by the **Z80-CTC** counter/timer circuit.

1.2 CENTRAL PROCESSING UNIT

The central processing unit is built with the 8-bit microprocessor **Z80A**.
Z80A is a **MOS-LSI** integrated circuit in a 40-pin **DIP** package, having 3 buses:

— the data bus;

— the address bus;

— the control bus.

The data bus **D0—D7** is a three-state input/output bus, used for exchanging information with the memory and the I/O interface circuits.

Z80A falls within the 8-bit microprocessor category, having the capability to process 8 bits of information simultaneously on its data bus.

The 16-bit address bus is used to address the memory or the I/O devices during the information exchanges.

Having 16 bits for the address bus, **Z80A** can address 64 KB of memory and an additional 64 KB space dedicated to the I/O devices.

The control bus offers the signals required for monitoring the data transfer to/from microprocessor.

The microprocessor can perform different tasks:

— reads data from memory;

— writes data to memory;

— reads data from I/O devices;

— writes data to I/O devices;

— performs arithmetic operations on data.

Z80A executes a range of 158 types of instructions. In the **CoBra** microcomputer, the microprocessor clock signal has a 3.5 MHz frequency.

Pin description:

A0-A15 — the address bus;
— three-state logic outputs, active in “1” logic level;
— can address up to 64 KB memory and I/O devices;
— in the case of I/O, the 8 less significant address bits are used to

select up to 256 input devices or 256 output devices;
— during the dynamic memory refresh cycle, the contents of the **I** and **R** registers shows up on the address bus, the 7 less significant bits of the self-incrementing **R** register being used as a refresh address.

D0-D7 — the data bus;
— three-state inputs/outputs, active in "1" logic level.

M1 — machine cycle one;
— output, active in "0" logic level;
— together with **MREQ**, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution;
— **M1** and **IORQ** both active indicate the execution of an interrupt cycle.

MREQ — memory access request;
— three-state output, active in "0" logic level;
— indicates that the address bus holds a valid address for a memory read of memory write.

IORQ — I/O ports access request;
— three-state output, active in "0" logic level;
— indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation;
— together with **M1**, indicates that an interrupt response vector can be placed on the data bus.

RD — read;
— three-state output, active in "0" logic level;
— indicates that the CPU wants to read data from memory or an I/O device.

WR — write;
— three-state output, active in "0" logic level;
— indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.

RFSH — refresh;
— output, active in "0" logic level;
— together with **MREQ**, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

HALT — halt state;
— output, active in "0" logic level;
— indicates that the CPU has executed a **HALT** instruction and is waiting for either a non-maskable or a maskable interrupt (with the mask enabled) before operation can resume. During **HALT**, the CPU executes **NOP**s to maintain memory refresh.

WAIT — wait;
— input, active in "0" logic level;

— indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer;
— the CPU continues to enter a **WAIT** state as long as this signal is active without refreshing the dynamic memory. Extended **WAIT** periods can prevent the CPU from properly refreshing dynamic memory.

INT — interrupt request;
— input, active in “0” logic level;
— interrupt request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled and **BUSRQ** is not active. **INT** is normally wired-OR and requires an external pull-up for these applications.

NMI — non-maskable interrupt;
— input, negative edge-triggered;
— **NMI** has a higher priority than **INT**. **NMI** is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

RESET — CPU reset;
— input, active in “0” logic level;
— initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state; dynamic memory refresh signals are not generated. Note that **RESET** must be active for a minimum of three full clock cycles before the reset operation is complete.

BUSRQ — bus request;
— input, active in “0” logic level;
— bus request has a higher priority than **NMI** and is always recognized at the end of the current machine cycle. **BUSRQ** forces the CPU address bus, data bus, and control signals **MREQ**, **IORQ**, **RD**, and **WR** to go to a high-impedance state so that other devices can control these lines. **BUSRQ** is normally wired-OR and requires an external pull-up for these applications. Extended **BUSREQ** periods due to extensive **DMA** operations can prevent the CPU from properly refreshing dynamic RAMs.

BUSACK — bus acknowledge;
— output, active in “0” logic level;
— indicates to the requesting device that the CPU address bus, data bus, and control signals **MREQ**, **IORQ**, **RD**, and **WR** have entered their high-impedance states. The external circuitry can now control these lines;
— as long as it is active, dynamic memory refresh signals are not generated.

1.3 MEMORY

The **Z80** microprocessor can directly access any location in a 64 KB memory block.

In order to have a maximum flexibility for applications, the **CoBra** microcomputer is bundled with the maximum **RAM** memory that can be accessed by the microprocessor, i.e. 64 KB. At power-up, the **RAM** memory has a random contents. Therefore the existence of a non-volatile memory is needed, one that wouldn't lose its contents at power-off (**EPROM**). The **CoBra** computer can be equipped with 2 to 16 KB of such a memory, the standard version having 2 KB.

In this memory there are recorded power-up specific programs, such as hardware tests, a test image used for adjusting the black & white or color monitor, computer configuration user selection indicating the source of the operating system to be loaded:

— **BASIC** interpreter – Sinclair ZX-Spectrum compatible, with a monitor program for work in assembler and with printing routines specific for **RS-232** serial interface printers;

— Sinclair ZX-Spectrum **BASIC** interpreter, unmodified, for applications that verify the genuineness of the existing ROM;

— operating system specialized in assembler work, with editor, assembler, disassembler, copier (ex. **OPUS**);

— any other user-designed interpreter. (ex. **FORTH**);

— **CP/M** compatible floppy disk operating system.

Spectrum-type operating system can be loaded from 16 KB **EPROM** memories, from magnetic tape or disk.

The **CP/M** operating system can only be loaded from disk, its operation involving the existence of the floppy disk.

On one hand, the Sinclair-Spectrum compatibility requires that the **RAM** memory between addresses **4000H** and **5AFFH** contain the information used by the video controller to display a 256x192 pixel image on the monitor, on the other hand running the **CP/M** operating system becomes impossible if the video memory shows up in the middle of the transient program area (**TPA**).

In order to solve this problem created by the dual nature of the computer, a memory configuration and selection circuit was created that satisfies the conditions imposed by these three configurations. This circuit is made of two D-type flip-flops (**U36**), a BCD-to-decimal decoder (**U56**) and gates, being presented in the block diagram in **Fig.2**.

The main advantage of this circuit is that it allows switching the memory configuration with a jump anywhere in the 64 KB space of the new configuration, even if the memory area containing the switching routine dissapears as a result of the switching.

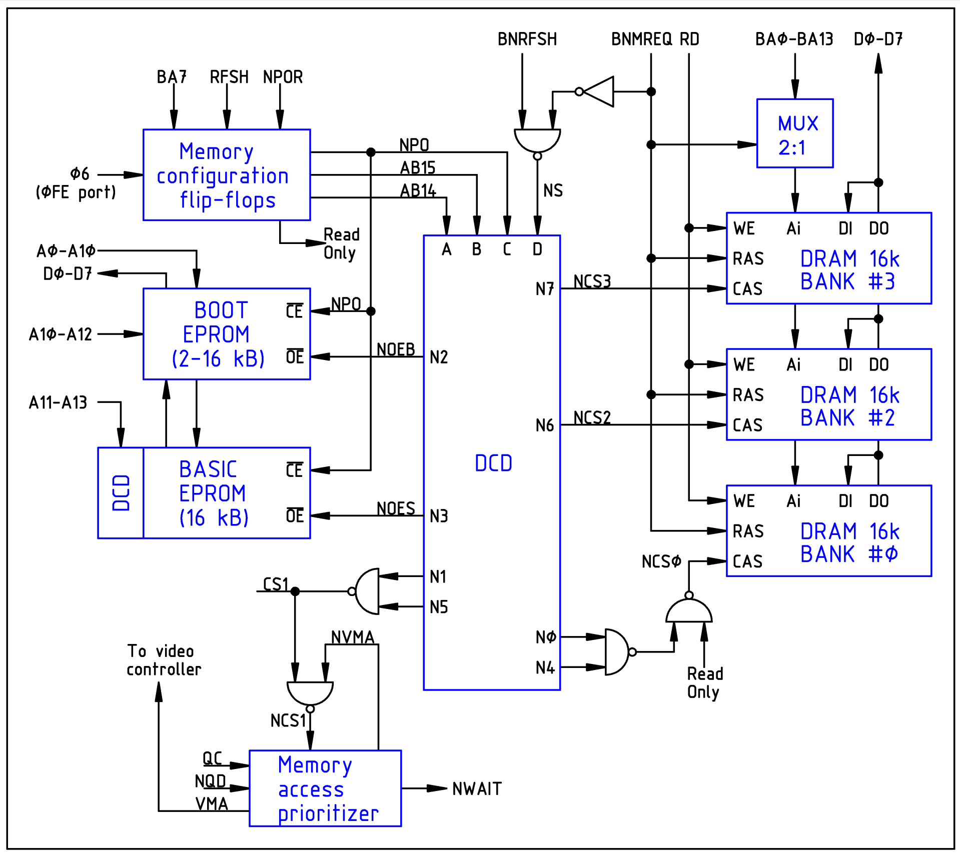


Fig. 2 - Memory configuration and selection circuit - block diagram

Its operation makes use of the fact that the **R** register is 7-bit incremented after each instruction fetch cycle and appears entirely (8-bit) on the address bus lines **A0—A7**, the moment of its appearance being signaled by the **RFSH** signal.

A change of bit 7 of the **R** register by the instruction **LD R,A** is shown on the address bus (**BA7**) only after the next instruction code was already fetched.

This instruction can be a single-byte jump instruction, such as **RST n** or **JP (HL)**, which can achieve the jump of the program counter to any address within the 64 KB memory space. The memory configuration switching sequence:

LD R,A ; JP (HL)

is detailed in **Fig.3**.

The two flip-flops (**U36**) are forced to “1” at power-up, by a circuit made of **C15**, **D02**, **R09**, **D03**.

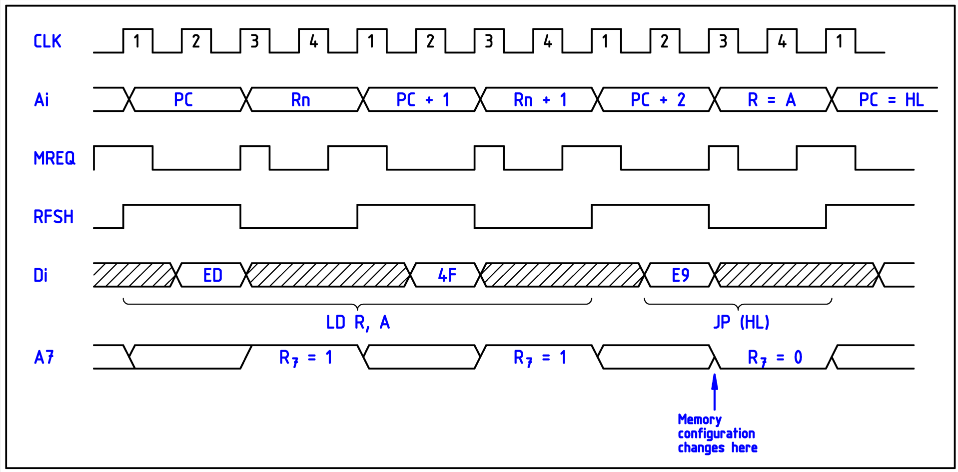


Fig. 3 - Memory configuration switching sequence

The diode **D02** makes sure the microprocessor exits the reset state a few miliseconds before the **NPOR** signal turns off. During this time the microprocessor executes the first instructions where bit 7 of the **R** register is set to “1”, which makes sure the flip-flop **U36/5** is maintained at “1” after **NPOR** turns off. This is the temporary startup configuration, where the memory map allows the microprocessor to access the **BOOT EPROM U89**, the 16k **BASIC EPROM**, the video memory and the 16k **DRAM** (**BANK #0**). In this startup configuration, the **RESET** button is not to be used, because the contents of the **R** register would be erased, which leads to a random memory configuration. The user can choose the configuration wanted by pressing one of the following keys:

- — B for BASIC interpreter resident in EPROM;
- — C for BASIC or other operating system resident on magnetic tape;
- — W to check the contact between the EPROM memories and their sockets. The contents of EPROMs are verified byte by byte against the recording on the magnetic tape. In case of error, the defective EPROM chip is indicated by flashing the corresponding color in the test image (0-7);
- — D to load the operating system from floppy disk.

After selecting the source of the operating system to be loaded from the keyboard, the **RAM** area between **8000H** and **FFFFH** is set up accordingly, bit 6 of the output port **FEH** is set to 0 or 1 according to the configuration chosen (**BASIC** or **CP/M** respectively) and bit 7 of the R register is brought back to “0”. The memory configuration will change after the microprocessor fetched the code for the **JP (HL)** instruction from the old configuration.

The map of the memory in these 3 configurations is shown in Fig. 4.

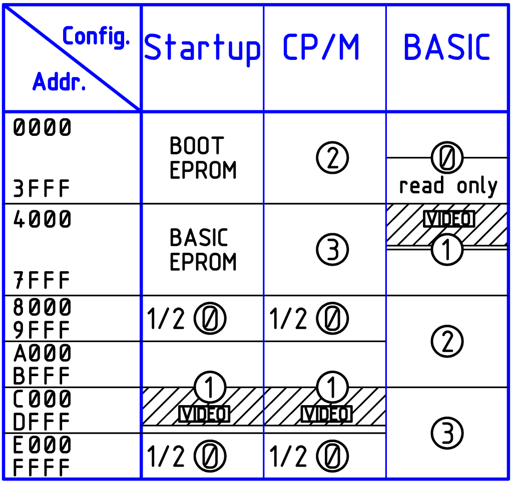


Fig. 4 - Memory map for the three configurations

Multiplexers **U41** and **U58** take care of changing the row and column address for the 48k **DRAM** memory (**U62-69**, **U43-U50**, **U24-U31**). In order to increase speed of operation in **CP/M**, half of the video **DRAM** (**BANK #1**) is replaced by half of **DRAM BANK #0**, otherwise the execution of **BDOS** calls would be slowed down by wait states from the memory access prioritizer. This memory replacement is achieved by the gates at **U52/12**, **U52/8**, **U35/3**.

Under the **BASIC** configuration, **U54/11** synthesizes the Read Only signal for **DRAM BANK #0**, the gate **U17/11** allows access for **SPECTRUM**-specific 20ms interrupts, and the asyncronous input **U36/1** locks the state of memory configuration flip-flops regardless of any change in bit 7 of the **R** register.

1.4 THE VIDEO CONTROLLER

The video image is organized in memory as follows:

— a 6kB area, called “serial video information area”, which specifies the type of every pixel of the screen, as follows:

if the corresponding bit is 0 then the pixel will have the color of “paper”, and if the bit is 1 then the pixel will have the color of “ink” for that particular character. The address of this 6kB area is **4000H** for **BASIC** configuration and **C000H** for the other two configurations;

— a 768 bytes area, called “color attribute area”, which specifies the color of “ink” and the color of “paper”, for each character, specifies if that character should be flashing and if it should have increased brightness. The address of this area is **5800H** for **BASIC** configuration and **D800H** for the other two configurations.

The block diagram of the video controller is shown in **Fig.5**.

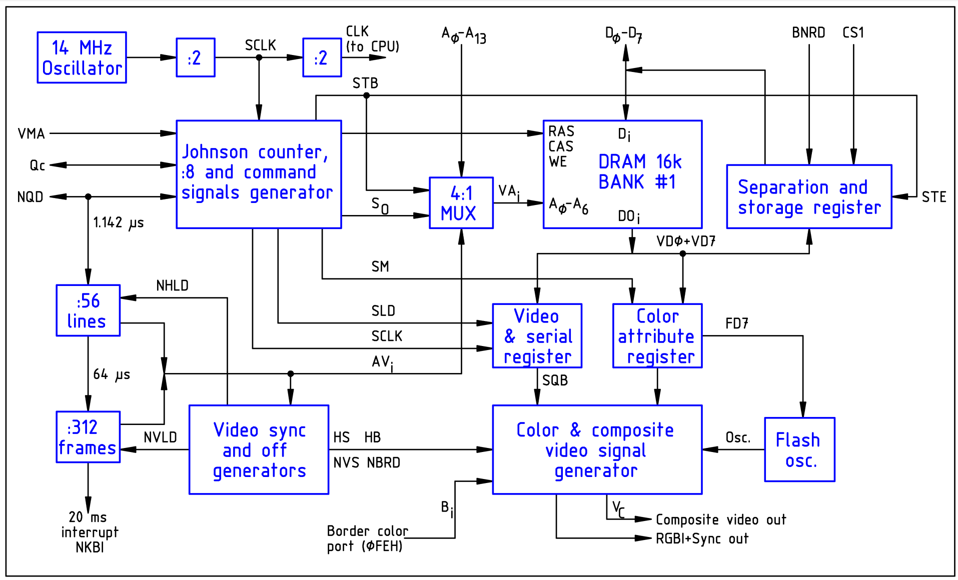


Fig. 5 - Video controller - block diagram

Starting from the 14 MHz clock built with a quartz oscillator with gates (**U57**), through a division by 2 the dot clock frequency **SCLK** is obtained - used for serialization - with a frequency of 7 MHz, at **U61**. This signal is further divided by 8 through the Johnson counter made with **U60** (**7495**), in order to generate the character frequency. Gate **U40/6** along with **U57/8** ensure the counter is self-triggered. The main advantage of this 4-bit divider by 8 is that the outputs' transitions occur one at a time, so at any given moment only one output changes its state. Through a simple decoding using gates, more signals are generated - control signals for **DRAM 16k BANK #1**, control signals for the serialization registers, the color attributes and the strobe for the data separation and storage register.

The timing of these signals and also the dual access to the video memory are both shown in **Fig.6**.

On one hand the video controller is accessing the video memory at fixed time intervals in order to read the serial video information and color attributes, on the other hand the CPU is accessing the video memory in order to change the image, the color attributes, the system variables or to store code or data.

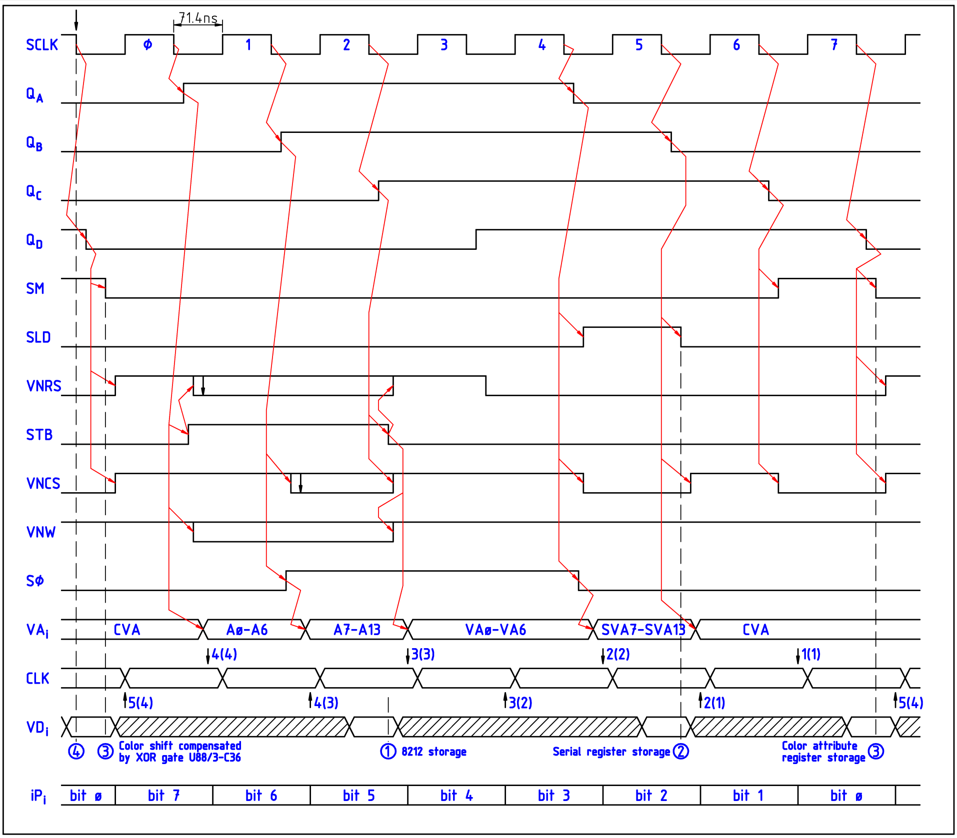


Fig. 6 - The timing of command signals

In order to easier calculate the address of any pixel as well as the address of the associated color attribute, we divide the 256x192 px image (or 32x24 characters) as follows:

— horizontally — 8 adjoined pixels, 8-bit encoded (**B7–B0**) make up one character;
— the image contains 32 characters, 5-bit encoded: **C4, C3, C2, C1, C0**;

— vertically — 8 sequential TV lines, 3-bit encoded **L2, L1, L0**, make up one character;
— 8 character rows, 3-bit encoded **R2, R1, R0** make up one third of the image;
— the image contains 3 thirds, 2-bit encoded **T1**, **T0**, the “11” combination not being used.

The address of the byte containing bit B of character C, line L, row R and third T can be found using formula **(1)**, and the address of the associated color attribute can be found using formula **(2)** where:

X=0 —> **BASIC** configuration

X=1 —> startup or **CP/M** configuration

```
A15 A14|A13 A12 A11 A10  A9  A8  A7| A6  A5  A4  A3  A2  A1  A0
 X   1 | 0   T1  T0  L2  L1  L0  R2| R1  R0  C4  C3  C2  C1  C0    (1)
       | +---------+ +---------+ +---------+ +-----------------+
       |     T           L         | R               C
       |                           |
 X   1 | 0    1   1   0  T1  T0  R2| R1  R0  C4  C3  C2  C1  C0    (2)
                         +-----+ +---------+ +-----------------+
                           T         R               C
```

In these formulas we notice the sameness of the **A0—A6** address bytes corresponding to the dynamic memory line address, which allows the video controller to access the memory for the two separate readings in page mode, **RAS CAS CAS**.

The multiplexing of the video addresses is done in two steps: circuits **U03**, **U20**, **U22**, **U39** do the multiplexing for **RAS-CAS** microprocessor and **RAS-CAS** video controller, and circuit **U51** along with gate **U19/8** do the address change for the two **CAS CAS** page-mode accesses of the video controller.

The access of the CPU to **DRAM BANK #1** (video) is less prioritized than the access of the video controller, and is controlled by the memory access prioritizer made with **U02/5** and **U02/9**. The access request is signaled by activating the **NCS1** line which in turn activates **NWAIT** through the asyncronous input **U02/1**. Possible moments of activation for the access request are marked on the **CLK** diagram of the CPU clock signal in **Fig.6** having beside them the number of wait states inserted by the memory access prioritizer in each case. The activation of the video memory access signal **VMA** leads to activating the **RAS**, **CAS**, **WE** signals; in case of reading, the byte requested is sampled and stored in the separation and storage register **i8212** (**U76**) — the moment marked as (1) in **Fig.6**. On the positive edge of the **QC** signal, **NWAIT** is deactivated, the microprocessor still maintaining the data read from memory on the data bus, by means of the **DS** selection signals of the **i8212** circuit.

The access of the video controller to **DRAM BANK #1** video is done once every 1.1µs, this way also ensuring the memory refresh.

At moment (2) in **Fig.6**, the data byte is loaded in the video & serialization register (**U78**, **U82**, **U83**), the role of **U83** being that of delaying the serial data by two **SCLK** periods before it reaches the select input of the **U80** multiplexer.

At moment (3), the color attribute byte is loaded into the color attribute register (**U77** and **U81**).

At moment (4) the video information (serialized) reaches the **QB** output (**U83/12**). The time difference between moments (3) and (4) can be compensated by the gate **U88/3** along with **C36**.

The color of ink or paper selected by **U80** is once again multiplexed with the border color, by **U85**.

Gate **U87/8** turns off the electron beam in the monitor during the horizontal-retrace. The outputs of **U85** are normalized in order to yield the intensity signal, and separated through **U86** to yield the **R**, **G**, **B**, **I** and sync signals for the color monitor. **U88/11** along with **T1** generates the **B/W** composite video signal.

The oscillator made of **U88/6**, **U88/8** is controlled by the **FD7** signal, together with gate **U88/3** achieving the **FLASH** function. Resistor **R43** ensures the blinking oscillator is in sync with the TV vertical sync signal. Its value must be chosen so that the oscillator flips during any TV line except the 192 ones that are visible.

The diode gates that synthesize the **BD6N** signal, along with **R73**, achieve the **BRIGHT** function, at the same time suppressing it for color black.

Circuits **U12** and **U13** and gates **U16**, **U14** make up a divider by 56. The outputs of this divider are being used as character address for the video controller and are the basis for horizontal sync and retrace signals. The associated signal timings are shown in **Fig.7**.

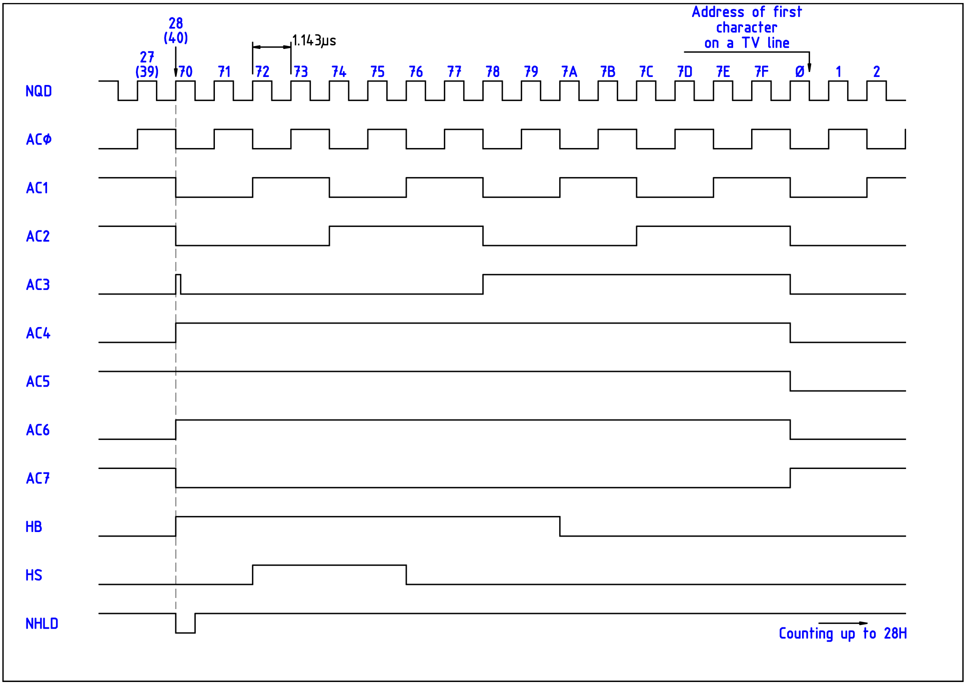


Fig. 7 - The timings for TV line signals

Circuits **U32**, **U33**, **U15/9** and gates **U37**, **U34** make up a divider by 312, their outputs being used as a TV line address, character row address and screen third address on one hand, and on the other hand they are the basis for generating the **NBRD** and **NVS** vertical sync signals.

The associated timings are shown in **Fig.8**.

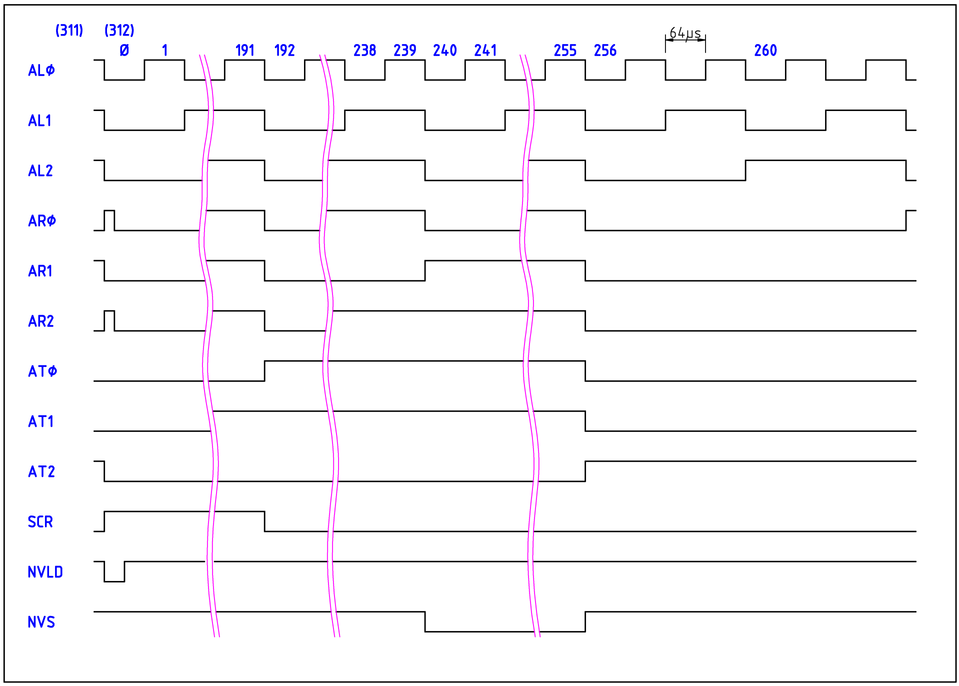


Fig. 8 - Timings for TV frame signals

2. MICROCOMPUTER INTERFACES

Gates **U54/6**, **U54/8** synthesize reading and writing signals from/to I/O ports. These signals are required for **INTEL** family interface circuits. The mainboard uses a programmable peripheral interface (**i8255**) which has three input/output ports and a control port as follows:

port **A** — input port, address 254 (**FEH**);

port **B** — input port, address 31 (**1FH**);

port **C** — output port, address 254 (**FEH**);

control port, address 223 (**DFH**), value 146 (**92H**).

∘ Bytes 0—5 of port **A** are used for reading the columns from the keyboard matrix.

∘ Bit **A6** is used for reading data from the external memory on magnetic tape. The audio signal from tape is limited by **R98**, **D10**, **D9** and formatted by **U92** (op. amp.).

∘ Bit **A7** is used as serial input, protected by **R94** and **D05**. It can be used as **RS-232** serial input when driven by a serial data reception routine (driver).

Input bits of port **B** can have a general use as an 8-bit parallel port at address 223 (**DFH**), with protocol signals **PA5** for input and **PC5** for output.

◦ Bits 0—4 of port **B** can be used as Kempston-compatible Joystick interface. Resistors **R99:106** ensure a "0" reading in stand-by, resistor **R107** ensure a "1" level through one of the joystick contacts.

◦ Bits 0—2 of port **C** are used for storing the **BORDER** color.

◦ Bit 3 serves as output for tape recording. Resistors **R97**, **R98** and diodes **D06**, **D07** ensure an optimum level for the majority of tape recorders.

◦ Bit 4 is used as audio output. The **i8255** circuit can directly drive a telephone speaker through **C45**, speaker mounted in the microcomputer's case.

◦ Bit 5 is a general use output bit. Can be used as a protocol line for port **B**.

◦ Bit 6 is an output line. It indicates the configuration selected at startup. In a particular configuration it can be used as a general use bit.

◦ Bit 7 is a serial output. It is separated and inverted by **U87/11**, the level being adapted by **T2** to be **RS-232C** compatible. It is used as a serial data transmission bit to a printer or another computer, using an emission routine.

The floppy disk interface is built on a separate board using the dedicated floppy disk controller circuit **i8272**.

To be able to use the features of **Mode 2** Interrupts of the microprocessor, the interrupts generated by **i8272** are passed through the counter/timer channels circuit **Z80-CTC** (**U01**). This circuit has four counters of which counters 0 to 2 are cascaded. By programming, channel 0 **CTC** generates one interrupt for each byte to be transferred between **i8272** and microprocessor; channels 1 and 2 cascaded generate an interrupt at the end of the sector also generating the Terminate Count (**TC**) signal for **i8272**. In this hardware configuration, **i8272** can be programmed to work without direct memory access, simplifying the interface very much.

The floppy disk interface block diagram is shown in **Fig.9**.

The clock signal generated by the 16 MHz quartz oscillator is divided by 2 or by 4 depending on the position of the switches **SD0-3** and is used as a clock for **i8272**. Remarkable is this original configuration where **i8272** - by means of the selection signals **US0**, **US1** - selects its own 8 or 4 MHz clock frequency for work with **8"** or **5¼"** floppy drives. This configuration allows the use of 2 floppy drives simultaneously regardless of their sizes, without having to intervene by means of software to switch the clock frequency to the floppy disk controller.

The circuit made with **U04/9** and **U03/4** performs the multiplexing for **READY** signals coming from floppy drives.

The double decoder circuit **U05** generates **SELECT** signals for floppy drives, as well as **HEAD LOAD** signals for the reading/writing heads of the drives.

Signals **HL0-3** (**HEAD LOAD**) can also be used as **MOTOR ON** signals for **5¼"** floppy drives which have this input line.

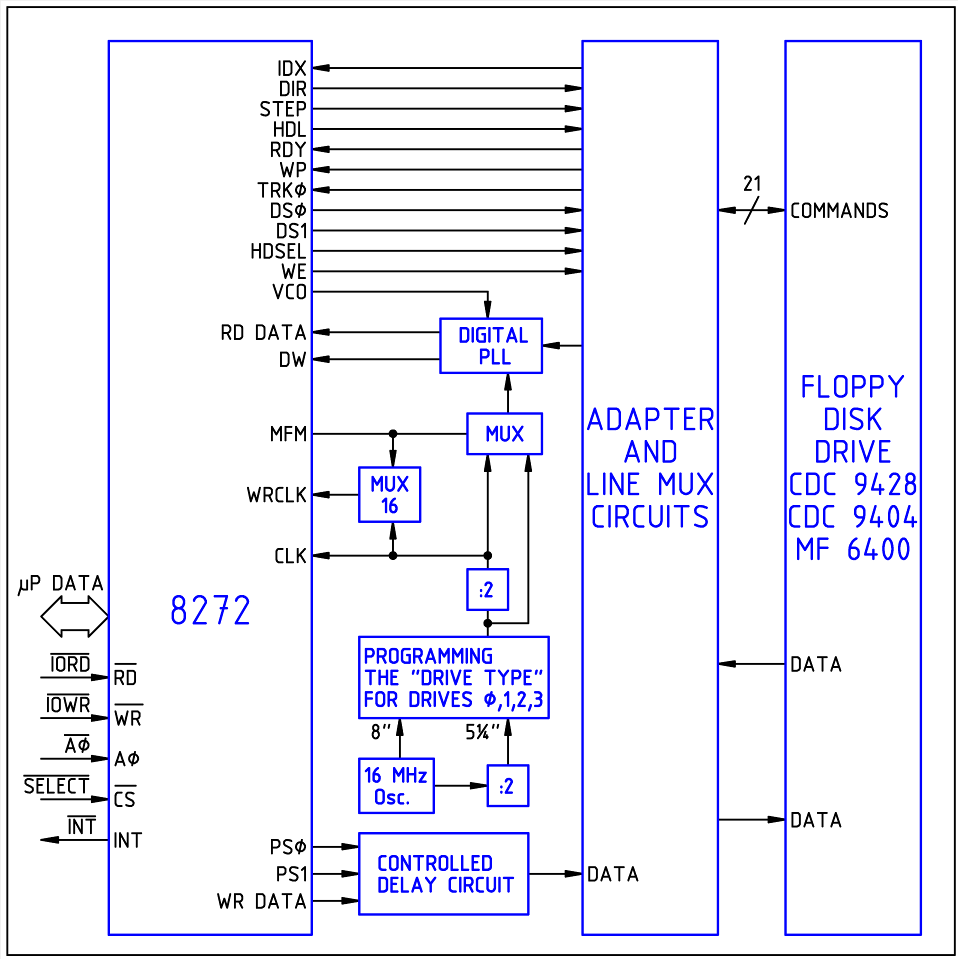


Fig. 9 - Floppy disk interface block diagram

In order to reduce the symbol crossover interference at write operations, a circuit made with **U18**, **U12** is used, that ensures precompensation for data to be written.

Gates **U16**, **U03**, **U06**, **U10** take care of multiplexing and level translation for command signals of the floppy drives.

Gates **U16**, **U15**, **U14/6**, **U14/8** and counter **U08** generate the write clock signal **WCK** and clock signal **CK** depending on the selected simple or double (**MFM**) density.

Circuits **U13**, **U07**, **U14/12**, **U09** make up a phase locked loop circuit (digital **PLL**) that is used to synthesize the data window **RDW** signal (Read Data Window) starting from the transitions of data received from the selected floppy drive. If input **U07/13** is in "1" logic level, **U13** along with **U07** make up a regular divider-by-16 circuit. The positive edge of the **USD** signal from the floppy drive

triggers a sampling of the current count of this divider. If the sample is 0, the frequency generated by the 16-divider is considered to be synchronous with data arriving from the floppy drive and the counter keeps counting. If the sample is not zero, a positive or negative jump by 1 or 2 is made in the counting sequence, according to the shift that was detected, so that the number in the 16-divider is brought closer to the correct synchronousness value. The **71488 PROM** is programmed as follows:

Address	Contents	Shift	Address	Contents
00	01	0	10	01
01	01	-1	11	02
02	02	-1	12	03
03	03	-1	13	04
04	03	-2	14	05
05	04	-2	15	06
06	05	-2	16	07
07	06	-2	17	08
08	0B	+2	18	09
09	0C	+2	19	0A
0A	0D	+2	1A	0B
0B	0E	+2	1B	0C
0C	0F	+2	1C	0D
0D	0F	+1	1D	0E
0E	00	+1	1E	0F
0F	01	+1	1F	00

Circuit **U09/6** performs a division by 2 on the output signal coming from the 16-divider, so that the **RDW** signal (data window) is generated from the clock signal **CK** through a division by 32 with a compensation of the shift occuring between **RDW** and the data read from the floppy disk.

3. KEYBOARD

The keyboard consists of 58 keys, of which 48 are organized as an 8x6 matrix, and the other 10 are used for generating some commands that on a standard **ZX SPECTRUM** are generated by simultaneously pressing **CAPS SHIFT** and another key.

The detection of a key press is done as follows: when interrogating the keyboard, the keyboard lines are connected to the higher 8 microprocessor address lines (**A8–A15**) separately through diodes **D11–D18**. During a keyboard reading cycle, these lines are sequentially brought to a “0” logical level, one by one, one at a time while the other 7 are in a “1” logical level. If a key is pressed, by the electrical contact made in that particular matrix node, the “0” level is propagated through the column (**k0–k6**) corresponding to the key that was pressed, all the way to the **PA** input port of the **i8255** circuit.

The keyboard schematics are shown in **Fig.10** and **Fig.11**.

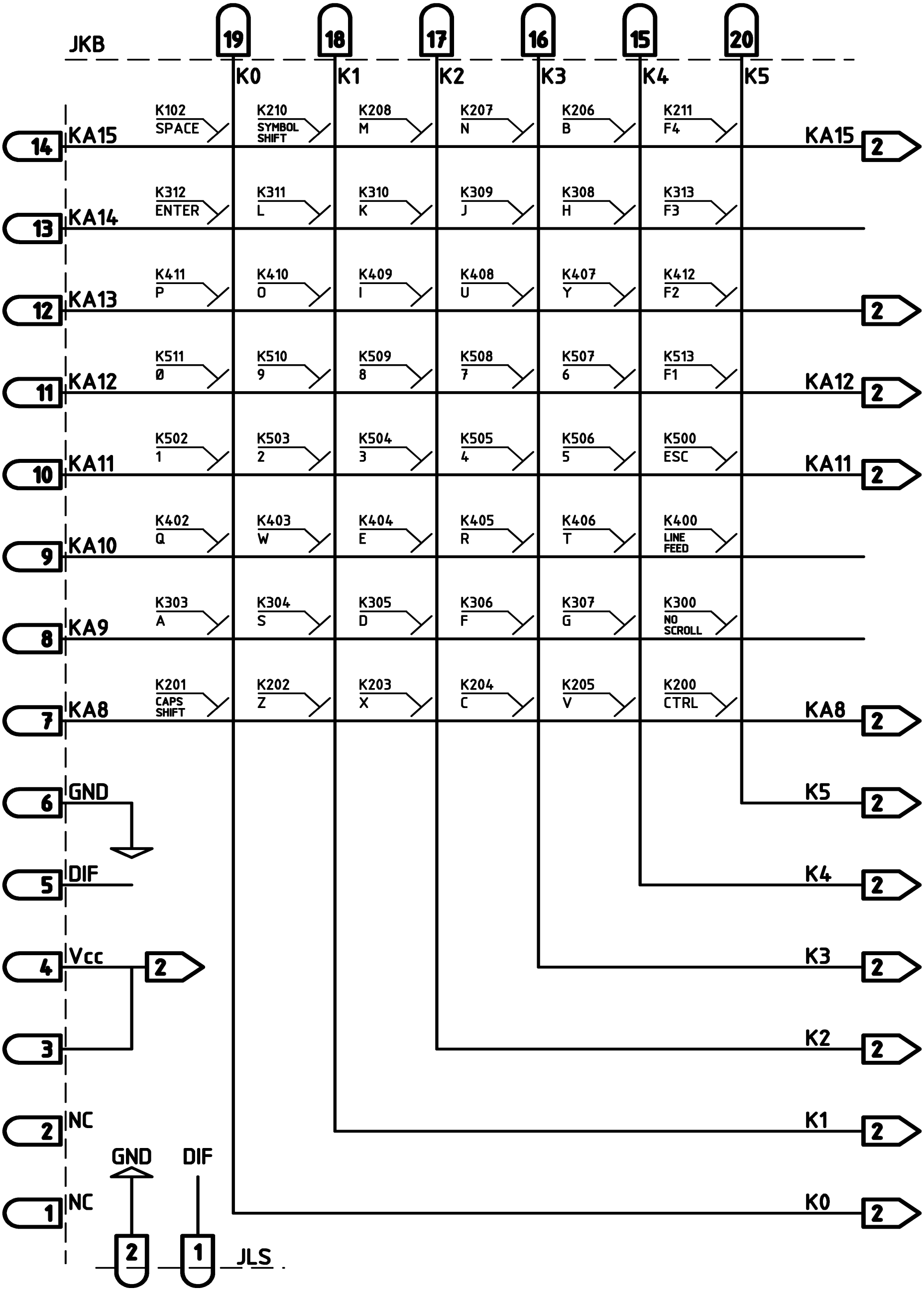


Fig. 10 - Keyboard schematic

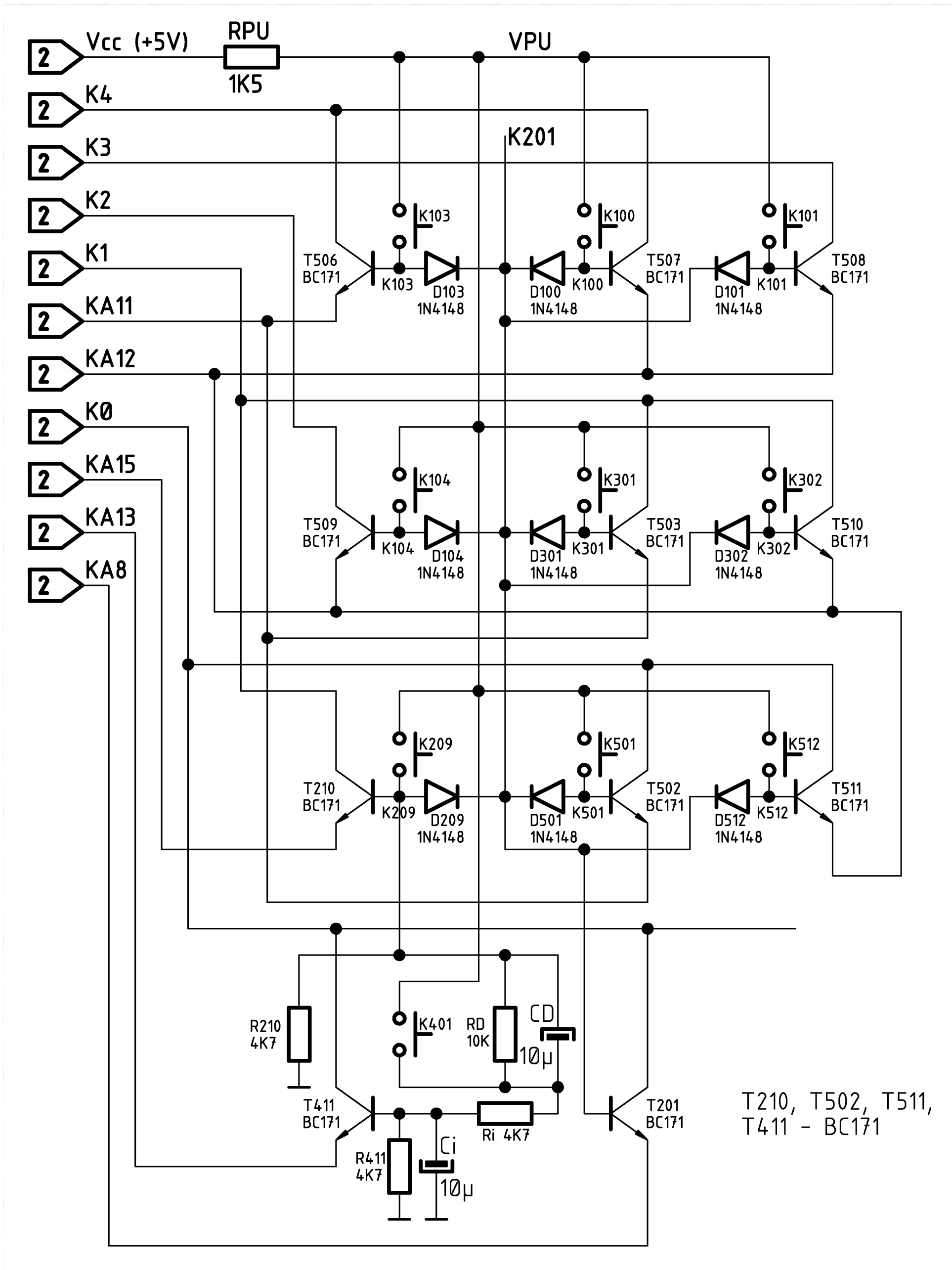


Fig. 11 - Keyboard schematic

The generation of a command with one of the 10 special keys mentioned above can be studied on the example in **Fig.12**; command **DEL** (deletion) which on **ZX SPECTRUM** is obtained by simultaneously pressing the **CAPS SHIFT** and **0** keys, on **CoBra** can be generated by pressing the **DEL** key.

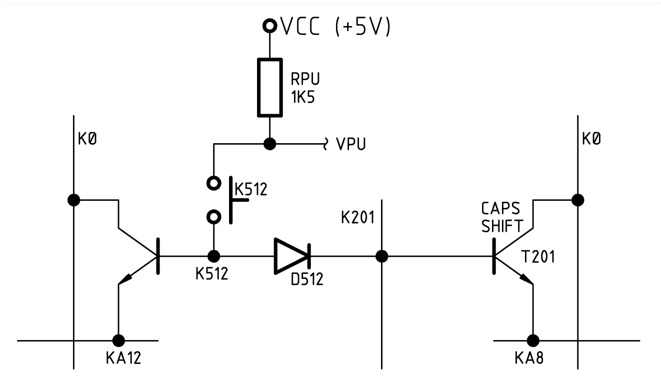


Fig. 12 - Special keys functional example

The **TAB** key in the group of 10 special keys is different from the others by the fact that it validates three different keys simultaneously: **CAPS SHIFT**, **SYMBOL SHIFT** and **P**. When pressing the **TAB** key, the contacts representing the **CAPS SHIFT** and **SYMBOL SHIFT** keys are activated, the command being propagated through the differential circuit **RD**, **CD**. The contact representing the **P** key does not yet get activated due to the presence of the integrating circuit **R411, CI**. Shortly after the deactivation of the contacts representing the **CAPS SHIFT** and **SYMBOL SHIFT** keys, the contact representing the **P** key is activated too.

4. POWER SOURCE

The power source of the **CoBra** computer can generate 3 different voltages:

+5V — for a load of max. 3A;

– 5V — for a load of max. 50mA;

+12V — for a load of max. 0.3A

The power source schematic is shown in Fig.13. The +5V voltage stabilizer is built as a switched-mode circuit. The main elements are:

— the group of T5, T4 (representing the switching transistor), IC3 (second generation stabilizer ROB 317);

— the group of L1, C15, which is the energy storage element;

— diode D5, which closes the load circuit during the off-time of the switching transistor;

— the group of R10, C14, R9.

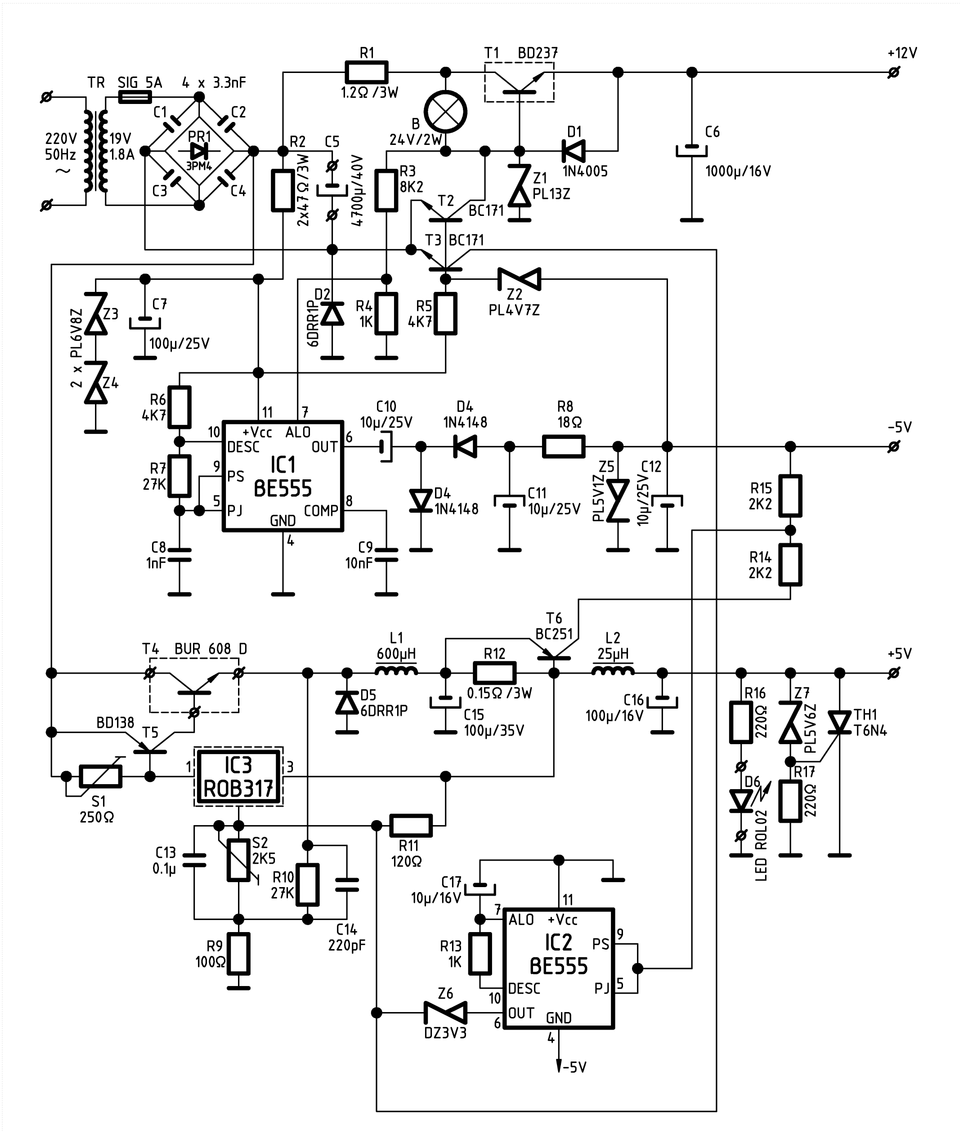


Fig. 13 - Power source schematic

At startup, the current through the load is supplied by the stabilizer circuit **IC3**. As the input current increases, the voltage on **S1** opens up the switching element made of **T5**, **T4** and the current through the coil starts rising in a linear fashion. The load current being constant, the current supplied by the linear stabilizer **IC3** decreases, as the current through **L1** increases. The process described eventually leads to the blockage of transistor **T5** and thus of the switching element **T5**, **T4**. At this time the voltage on coil **L1** changes polarity, diode **D5** opens up and the energy stored in coil **L1** supplies the load current during the on-time of **T5**, **T4**. After a while, the output voltage tends to decrease below the preset value, this being sensed by the **IC3** stabilizer which initiates a new conduction process in the **T5**, **T4** transistors. The positive feedback loop **R10**, **C14**, **R9** introduces a hysteresis in the voltage applied on the linear stabilizer input, helping its turning on or off.

The +5V voltage is adjusted with the trimmer resistor **S2**.

The optimal efficiency is adjusted with the trimmer resistor **S1**.

The –5V voltage stabilizer is a **DC–DC** type. It is made up of an oscillator built with **IC1** (**ßE 555**), **R6**, **R7** and **C8**, a voltage doubler rectifier circuit built with **C10**, **D3**, **D4** and **C11**, followed by a shunt regulator built with **R8**, **Z5** and the filter capacitor **C12**.

The oscillator is fed on pin 11 with a voltage of about 14V obtained from the regulator built with **R2**, **Z3**, **Z4** and the filter capacitor **C7**. On output pin 6 of **IC1** a rectangular signal is generated with a frequency f=20kHz and a duty cycle close to 1/2.

The +12V voltage stabilizer is an emitter follower transistor regulator. It is built with **R1**, **B**, **T1**, **Z1** and the filter capacitor **C6**.

Restrictions and protections:

To avoid damaging the RAM memory it is necessary that the first voltage applied to the memory be the –5V voltage. The power source of the **CoBra** microcomputer achieves this condition with the help of the **R5**, **Z2** group and the **T2**, **T3** transistors. At source power-up, the output of the –5V stabilizer is at 0V. Diode **Z2** is blocked and **T2**, **T3** are saturated due to the current being injected in their bases through **R5**. The negative potential from the cathode of diode **D2** appears on the collectors of transistors **T2**, **T3** and thus also on the base of **T1** and pin 2 (reference) of **IC3**. The two voltage stabilizers (+12V and +5V) will be as such kept at a positive value close to 0V. When the –5V power source output gets close to the preset value, diode **Z2** opens up and pulling the voltage potential on the bases of transistors **T2**, **T3** down blocks them, allowing the +12V and +5V sources to be turned on. The +5V source is designed with a protection against exceeding a 3A output current. This protection works as follows:

Increasing the output current over 3A causes a voltage on **R12** equal to the voltage required to open the base-emitter junction of transistor **T6**. The positive potential which appears on the collector of **T6**, divided by **R14**, **R15** and applied to pins 5, 9 of **IC2**, is bigger than the upper trigger limit of **IC2**, which works as a flip-flop circuit.

IC2, which has the output pin 6 in a high state (0V) at startup (power-on), flips to a low state (–5V). The Zener diode **Z6** opens up holding the potential of pin 2 of **IC3** relative to ground at approx. –1.2V. The output voltage will be equal to the sum of the voltage on **S2**, **R9** and the voltage on **R11** (1.2V — datasheet value), that is approx. 0V. Rearming the +5V voltage can be done by turning the main switch off and then on. Due to the tight admissible tolerances for the +5V voltage (±0.25V), in order to avoid accidental increases of the +5V voltage, a crowbar protection was also designed, built with **TH1**, **Z7**, **R17**, which upon exceeding a voltage of approx. 6V triggers the overcurrent protection by opening **TH1**.

The protection on the +12V source works as follows: when a shortcircuit on +12V occurs, on the base of transistor **T1** a positive potential less than 1V relative to ground appears, which divided by **R3**, **R4** is applied to pin 7 (**ALO**) of **IC1**, blocking the oscillator. Thus the –5V voltage dissapears, **Z2** is blocked, transistors **T2**, **T3** are opened and through **T2** saturated the oscillator built with **IC1** is held in a blocked state. The light bulb **B** was included in the +12V stabilizer schematic due to the nonlinear character of its resistance, this way achieving a limitation of the shortcircuit current by limiting the base current for transistor **T1**. Through **T3** saturated, the +5V voltage will be close to 0V too, so in conclusion a shortcircuit on the +12V output will cut all three voltages off. rearming is done by turning the main switch off and then on.

The protection on the –5V source works as follows: a shortcircuit on the –5V output causes **Z2** to block, **T2**, **T3** get saturated and through **T2**, **R3**, **R4** the potential on pin 7 of **IC1** is close to 0V, blocking the oscillator built with **IC1**. It is obvious that this state is maintained until the source is rearmed. All the three stabilizers are blocked until turning the main switch off and then on.

Powering up:

After verifying the connections, one pin of diode **Z7** is disconnected to avoid triggering the crowbar protection, **S1** is positioned in the middle and **S2** set to its minimal resistance, and then the circuit is plugged in. The presence of the +5V voltage is verified. If it is not present, the oscillator built with **IC1** should be checked. After this, the presence of the +12V voltage is verified after which the +5V voltage is adjusted without load. The oscilloscope is connected between the cathode of **D5** and ground. The system should be oscillating.

After this, a load of approx. 1.3 Ohms and at least 12W is connected to +5V, a load of 39 Ohms/5W connected to +12V and a load of 100 Ohms/0.5W connected to –5V.

All voltages are again checked, adjusting the +5V voltage with S2. The voltage on **D5** is visualized on the oscilloscope checking that the frequence is around 33 kHz (+2 kHz, – 4 kHz) and the amplitude around 18V.

A fine adjusting of the frequence can also be done by modifying the resistance of **R10**.

In order to get to the maximum efficiency, the fuse is pulled out of the socket on the PCB, inserting instead an AC ammeter (**MAVO-35** set to the 5A~ position). The main switch is turned on and **S1** is carefully adjusted until a low

is found. Then the +5V voltage is checked again and adjusted.

The shortcircuit protections on +12V and –5V are tested, respectively the triggering of the +5V protection against a load current in excess of 3A. **Z7** is then connected and the +5V voltage is increased by adjusting **S2** until **TH1** is activated. After this last check the +5V voltage is readjusted. After this, the power source is ready to be used.

WARNING!

Do not insert and do not pull out the power connector (**j9**) while the power source is on; the memory chips with three voltages (**4116**) might get destroyed.

5. COMPONENT LIST - FLOPPY DISK INTERFACE

CODE	TYPE
u01	Z80A-CTC
u02	i8272A
u03	74(LS)14/7404
u04	74(LS)153
u05	74156/74155
u06	7438/7403
u07	74S174
u09	74(LS)74
u10	74(LS)08
sw	dip-switch
u12	74(LS)153
u13	74(S)188
u14	74(LS)10
u15	74(LS)51
u16	74(LS)04
u17	74(LS)74
u18	7495
xtal	16 MHz
R01	150/0.5W
R02	150/0.5W
R03	150/0.5W
R04	150/0.5W
R05	1K/0.5W
R06	150/0.5W
R07	150/0.5W
R08	150/0.5W
R09	150/0.5W

R10	470/0.5W
R11	470/0.5W
R12	470/0.5W
R13	470/0.5W
R14	680/0.5W
R15	680/0.5W
R16	4K7/0.5W
c01	1µ/35V
c02	100n/50V
c03	1µ/35V
c04	100n/50V
c05	220p/30V
c06	220p/30V
jex	201.607 RC
jfdd	201.243 RC
pcb	CX-FDC PCB rel 2.0 921442432

6. COMPONENT LIST - COBRA MICROCOMPUTER

CODE	TYPE	PCS.
u01, u18, u23, u42, u59, u75, u84, u91	2716 (2732)	8(4)
u02, u15, u36, u61	74(LS)74	4
u03, u20, u22, u39	74(LS)153	4
u[04:11], u[24:31], u[43:50], u[62:69]	4116 (4516)	32

u12, u13, u32, u33	74(LS)193	4
u14, u40, u52	74(LS)10	3
u16, u17, u37, u54, u87, u55	74(LS)00	6
u19, u38	74(LS)08	2
u21	74(LS)51	1
u34	74(LS)20	1
u35, u88	74(LS)86	2
u41, u58	74157	2
u51, u80, u85	74(LS)157	3
u53, u57	74(LS)04	2
u56, u70	74(LS)42	2
u60, u[77:78], u[81:83]	74(LS)95	6
u[71:74], u86	74(LS)07	5
u76	i8212	1
u79	i8255	1

u89	2716 (2732) (2764) (27128)	1
u90	Z80A-CPU	1
u92	741	1
xtal	14 MHz	1
t1	2N2222	1
t2	BC251	1
p[01:04], p[06:07], p[09:18]	1N4148	16
p05, p08	DZ4V7	2
ls	phone speaker	1
j1	300.066	1
j2	300.064 300.060	1
j3, j4, j9	303.608A	3
j5	201.577	1
j6	300.062	1
j7	201.561	1
j8	201.161 RC+	1
	201.146 RC	1
jex	201.619 RC	1
k1	EA-5993	1
sb	SWITCH	1
ss	DIP-SWITCH	1

R[01:05], R[07:08], R[18:19], R21, R[24:25], R[28:30], R[32:33], R[36:37], R[39:41]	33	22
R06, R20, R22, R23, R26, R27, R31, R34, R42, R73, R94, R96	1K	12
R[99:106]	8K2	8
R09, R84, R87, R107	2K2	4
R[10:17], R69, R85, R[108:113]	4K7	16
R35, R38	680	2
R43	1–4K7*	1
R44	330–390	1
R[45:47], R[49:66]	620	21
R48, R92	150/0.5W	2
R70	330/0.5W	1

R[67:68], R[89:90], R95, R[97:98]	10K	7
R[71:72], R[75:77], R91	100/0.5W	5
R72, R[78:83]	470	7
R73	1K	1
R74,RN	240	1
R86	220/0.5W	1
R93	75/0.5W	1
Rw	4K7	1
c[01:05], c[07:13], c[16:23], c[25:29], c[31:32], c[37:39], c41, c[46:48]	100n/30V	34
c14, ccS, cAT, cHB	150pF	4
c15, c45	10µ/10V	2
c24	220pF	1
c30	15pF	1
c[33:35], c40	1–10µ/35V	4
c36	270pF	1
c42	100–220µ/6V	1
c43	100–330µ/6V	1
c44	150–330µ/10V	1
cXR, ccc	390pF*	2
pcb	921442431	1

7. COMPONENT LIST - KEYBOARD

CODE	TYPE	PCS.
t201, t210, t411, t[502:503], t[506:511]	BC171	11
d[100:101], d[103:104], d209, d[301:302], d501, d512	1N4148	9
cd, ci	10μ/35V	2
r201, r411, ri	4K7/0.125W	3
rd	10K/0.125W	1
rpu	1K5/0.125W	1
jkb	201.606	1
	369.108.151	116
	369.108.152	58
	369.108.154	58
	369.108.155	58
	369.108.169	2
	369.108.170	1
k[100:101], k[103:104], k200, k209, k211, k[300:302], k313	369.108.153y	11
k102	369.108.159x	1
k201, k210	369.108.158x	2

k[202:208], k[303:311]	369.108.153x	16
k312	369.108.163x	1
k400, k412	369.108.156y	2
k401	369.108.162y	1
k[402:411]	369.108.156x	10
k500, k513	369.108.157y	2
k501, k512	369.108.164y	2
k[502:511]	369.108.157x	10

8. COMPONENT LIST - POWER SOURCE

CODE	TYPE
CI1	ßE 555
CI2	ßE 555
CI3	ROB 317 (TO-39)
T1	BD 237
T2,T3	BC 171
T4	BUR 608
T5	BD 138
T6	BD 251
PR1	3PM05
D1	1N4001
D2	6DRR1P
D3,D4	1N4148
D5	6DRR1P
D6	LED ROL 02
Z1	PL 13Z
Z2	PL 4V7Z
Z3,Z4	PL 6V8Z
Z5	PL 5V1Z
Z6	DZ 3V3
Z7	PL 5V6Z
C1,C2,C3,C4	3n3 (CLX 12.15)
C5	4700µF/40Vcc (EG 76.91)
C6	1000µF/16Vcc (EG 61.44)
C7	10µF/25Vcc (EG 52.53)
C8	1nF (CLY 12.06)
C9	10nF (CLX 12.15)
C10,C11,C12	10µF/25Vcc (EG 52.44)

C13	100nF (CLY 32.15)
C14	220pF (CLZ 12.06)
C15	100µF/35Vcc (CTS-M)
C16	100µF/16Vcc (EG 61.21)
C17	10µF/16Vcc (EG 52.43)
R1	1.2/3W (RBC-1002)
R2	2x47/3W (RBC-1002)
R3	8K2 (RCG 10.50)
R4	1K (RCG 10.50)
R5,R6	4K7 (RCG 10.50)
R7	27K (RCG 10.50)
R8	18K (RCG 10.50)
R9	100 (RCG 10.50)
R10	27K (RCG 10.50)
R11	120 (RCG 10.50)
R12	0.15 (RBC-1002)
R13	1K (RCG 10.50)
R14,R15	2K2 (RCG 10.50)
R16	220 (RCG 10.50)
R17	270 (RCG 10.50)
S1	250 (P32824)
S2	2K5 (P32824)

Other materials:

— TR 220V/19V S=8 cm^2
N2=96 turns 0.95mm enamelled copper (≈19AWG)
N1=1067 turns 0.2mm enamelled copper (≈32AWG)
— DIN 5 pin connector (303608)
— B — auto light bulb 24V/2W
— Fuse socket for PCB mount
— L1=600µH (1.2mm enamelled copper (≈17AWG) on 34x28mm ferrite pot core)
Ex: AL=570nH/turn : n=33 turns.
— L2=25µH (1.2mm enamelled copper (≈17AWG) on 23x17mm ferrite pot core)
Ex: AL=280nH/turn : n=10 turns.

CONNECTORS – pin-signal listing

```
j1 – video connector

      1 – VVM    – modulator power
      2 – GND    – ground
      3 – VR     – color red
      4 – VG     – color green
      5 – VB     – color blue
      6 – VI     – brightness
      7 – VC     – monochrome composite video signal
      8 – VS     – positive synchronization signal
      9 – VNS    – negative synchronization signal


j2 – RS 232 connector

      2 – TxD
      3 – RxD
      7 – GND


j3 – auxiliary connector

      1,4 – AA   – speaker out
      2   – GND  – ground
      3,5 – ATR  – formatted audio signal


j4 – tape recorder connector

      1,4 – ATO  – output to tape recorder
      2   – GND  – ground
      3,5 – ATI  – input from tape recorder


j5 – 8 bit input port connector address 0DFH

       1a – 8a   – GND   – ground
       9a – 10a  – Vcc   – +5V
       1b        – PB0
       2b        – PB1
       3b        – PB2
       4b        – PB3
       5b        – PB4
       6b        – PB5
       7b        – PB6
       8b        – PB7
       9b        – K5    – bit 5 port 0FEH – input
      10b        – 05    – bit 5 port 0FEH – output
```

j6 – Joystick connector

```
 1    – PB0 – right
 2    – PB1 – left
 3    – PB2 – down
 4    – PB3 – up
 5    – PB4 – button[1]
 6          – NC    [2]
 7          – NC
 8    – JSC – common[3]
 9          – NC
10          – NC    [4]
```

– 1) The original PCB layout had PB4 connected to j6/pin5 but the original manual had PB4 listed at j6/pin6. I kept the original PCB layout and I changed this manual to match the PCB.
– 2) Same as above.
– 3) The original PCB layout had JSC connected to j6/pin10 but the original manual had JSC listed at j6/pin8. I changed the PCB layout to match the original manual (maintained above).
– 4) The original manual only had 9 pins listed for the Joystick connector. The original PCB layout though had 10 holes drilled for this connector and pin 10 was even connected to JSC as mentioned above at 3). I changed the original manual adding pin 10 to the list, even though it is now not connected.

j7 – keyboard connector

```
 1a   – NC
 2a   – Vcc – +5V
 3a   – DIF – phone speaker out
 4a   – KA8 – A8 – keyboard
 5a   – KA10 – A10
 6a   – KA12 – A12
 7a   – KA14 – A14
 8a   – K4 – bit 4 port 0FEH – input
 9a   – K2 – bit 2
10a   – K0 – bit 0
 1b   – NC
 2b   – VCC – +5V
 3b   – GND – ground
 4b   – KA9 – A9 keyboard
 5b   – KA11 – A11 keyboard
 6b   – KA13 – A13 keyboard
 7b   – KA15 – A15 keyboard
 8b   – K3 – bit 3
 9b   – K1 – bit 1
10b   – K5 – bit 5
```

j8 – expansion connector

	a	b	c
1	VDD	VDD	VDD
2	GND	GND	GND
3	VBB	VBB	VBB
4	VCC	VCC	VCC
5	BA15	–	BA14
6	BA7	–	BA8
7	BA6	–	BA9
8	BA5	–	BA11
9	BA4	–	–
10	BA3	–	BA10
11	BA2	–	NPO
12	BA1	–	D7
13	BA0	–	D6
14	D0	–	D5
15	D1	–	D4
16	D2	–	D3
17	BA12	–	BA13
18	–	–	–
19	–	–	–
20	06	–	05
21	IEI	–	–
22	IEO	–	–
23	NIOWR	–	NIORD
24	BNRD	–	BNIORQ
25	BNWR	–	BNMREQ
26	BNRFSH	–	BNM1
27	NHALT	–	NWAIT
28	NBUSACK	–	NNMI
29	NRST	–	NBUSRQ
30	POR	–	NINT
31	GND	–	GND
32	BCLK	BCLK	BCLK

j9 – power connector

1	VDD +12V
2	GND ground
3	VCC +5V
4	VBB -5V
5	GND ground

jEX – floppy disk interface connector

	a	b
1	VCC	VCC
2	D3	D4
3	D2	D5
4	D1	D6
5	D0	D7
6	–	–
7	POR	BNRD
8	BA1	NIORD
9	SI/TRG3	NIOWR
10	BA4	BNIORQ
11	BA3	IEO
12	NRST	NINT
13	BA2	IEI
14	BCLK	BNM1
15	GND	GND

jFDD – floppy disk drive connector

1a	NSS
2:20a	GND
1b	NUSD
2b	NTR00
3b	NWP
4b	NIX
5b	NRDY0
6b	NRDY1
7b	NRDY2
8b	NRDY3
9b	NS0
10b	NS1
11b	NS2
12b	NS3
13b	NHL3
14b	NHL2
15b	NHL1
16b	NHL0
17b	NWD
18b	NST
19b	NDIR
20b	NWG

9. APPENDIX 1

CoBra Microcomputer Schematics

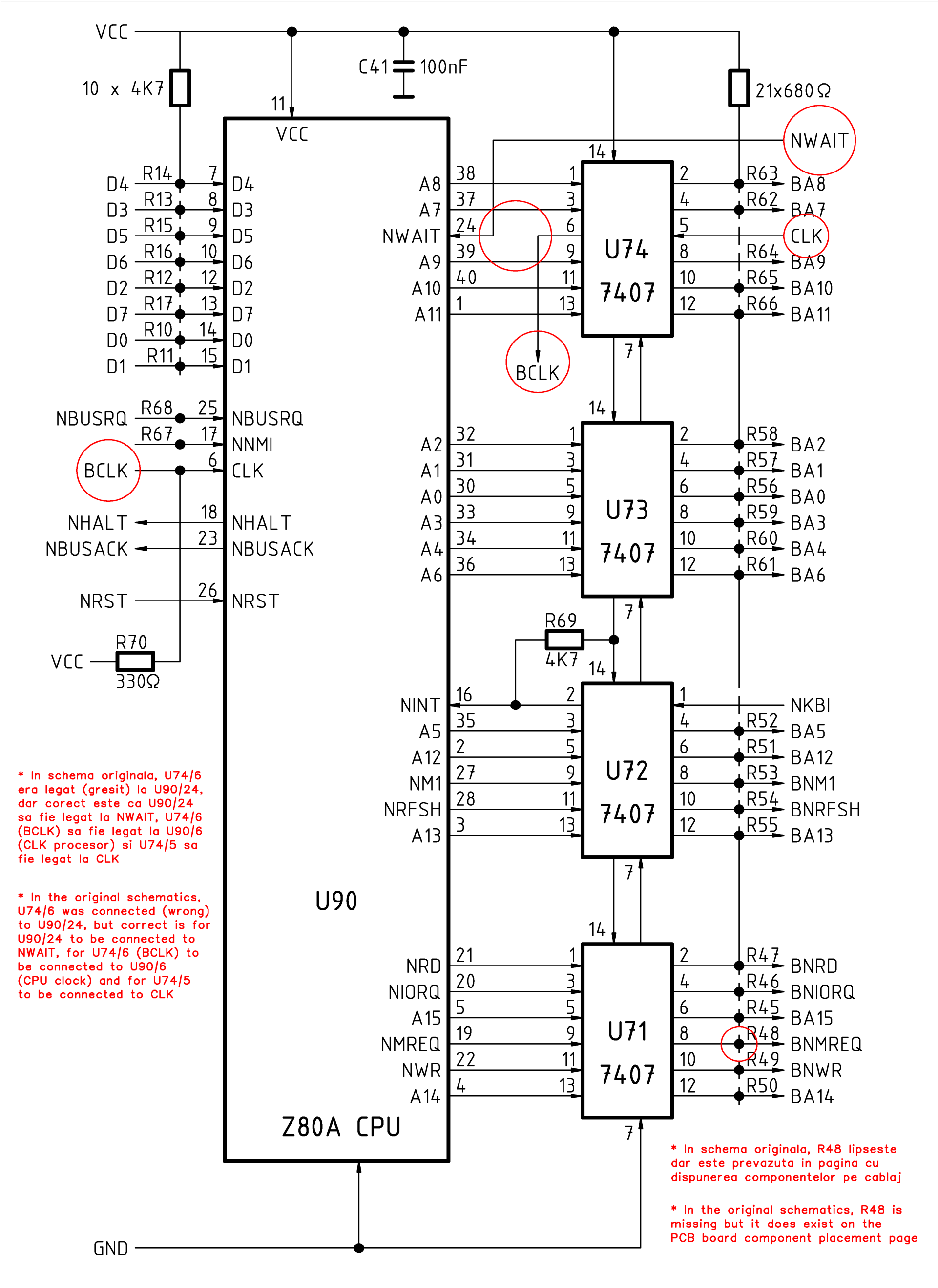


Fig. A1.1 – CoBra Microcomputer. Central Processing Unit.

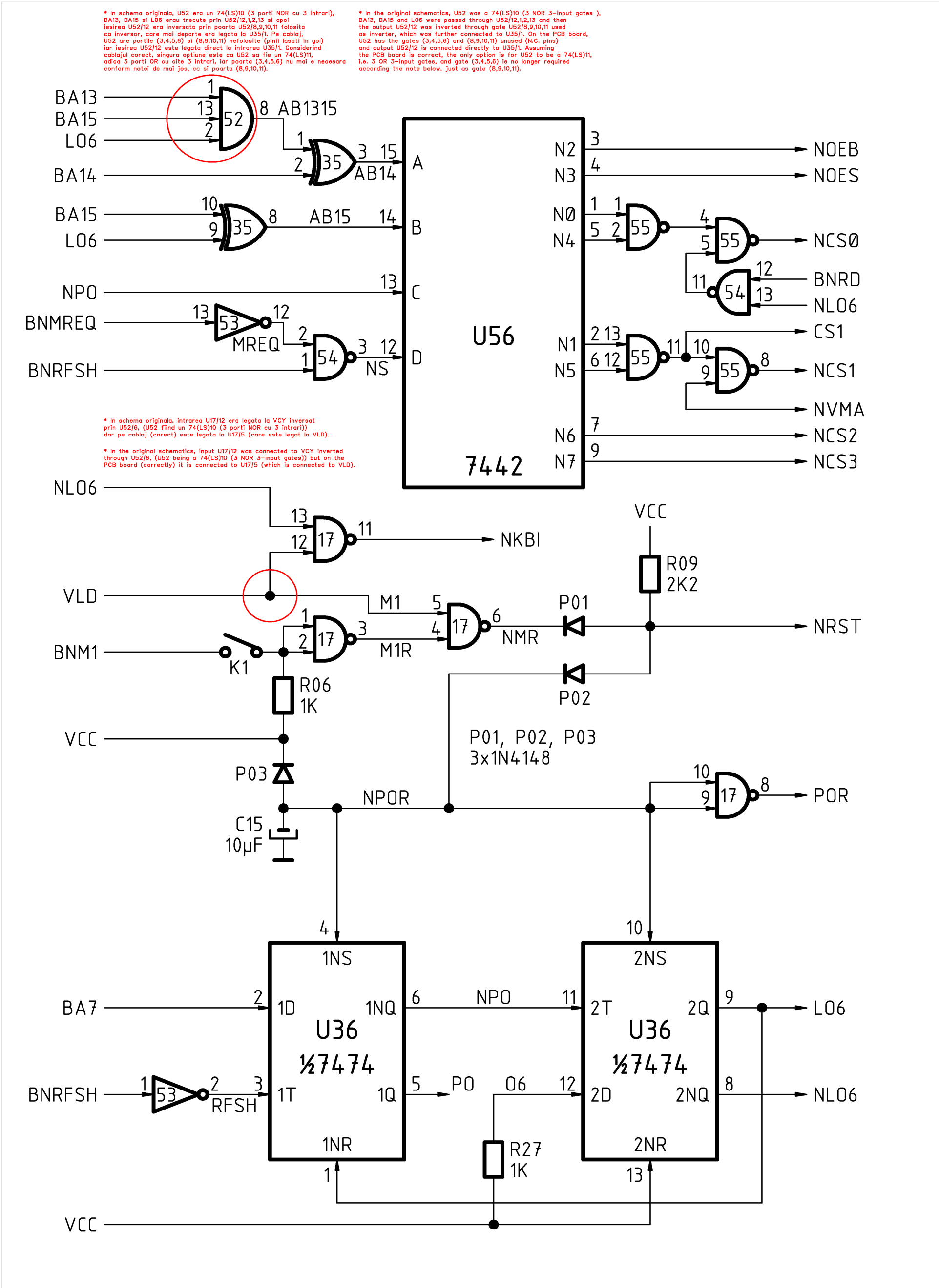


Fig. A1.2 – CoBra Microcomputer. Configurator and Selector Circuit.

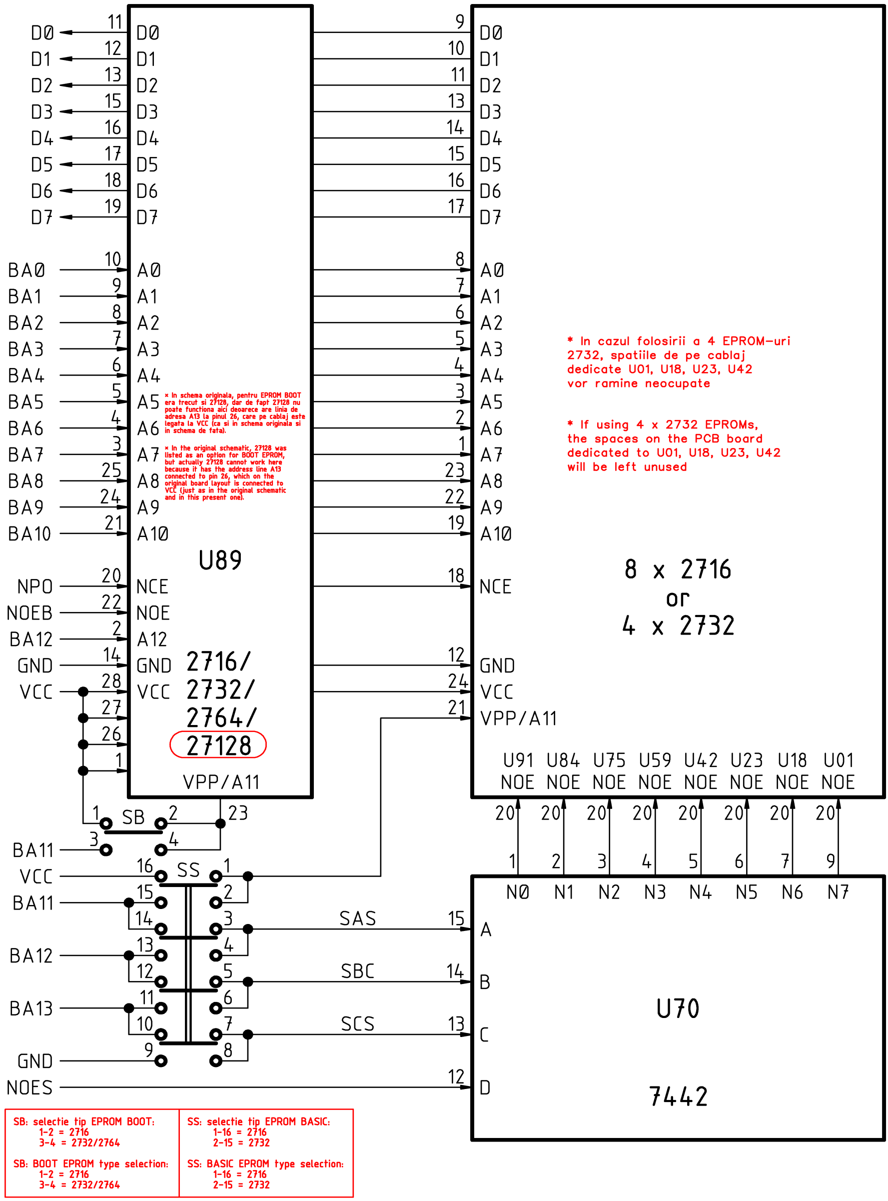


Fig. A1.3 – CoBra Microcomputer. Read-Only Memory Circuit.

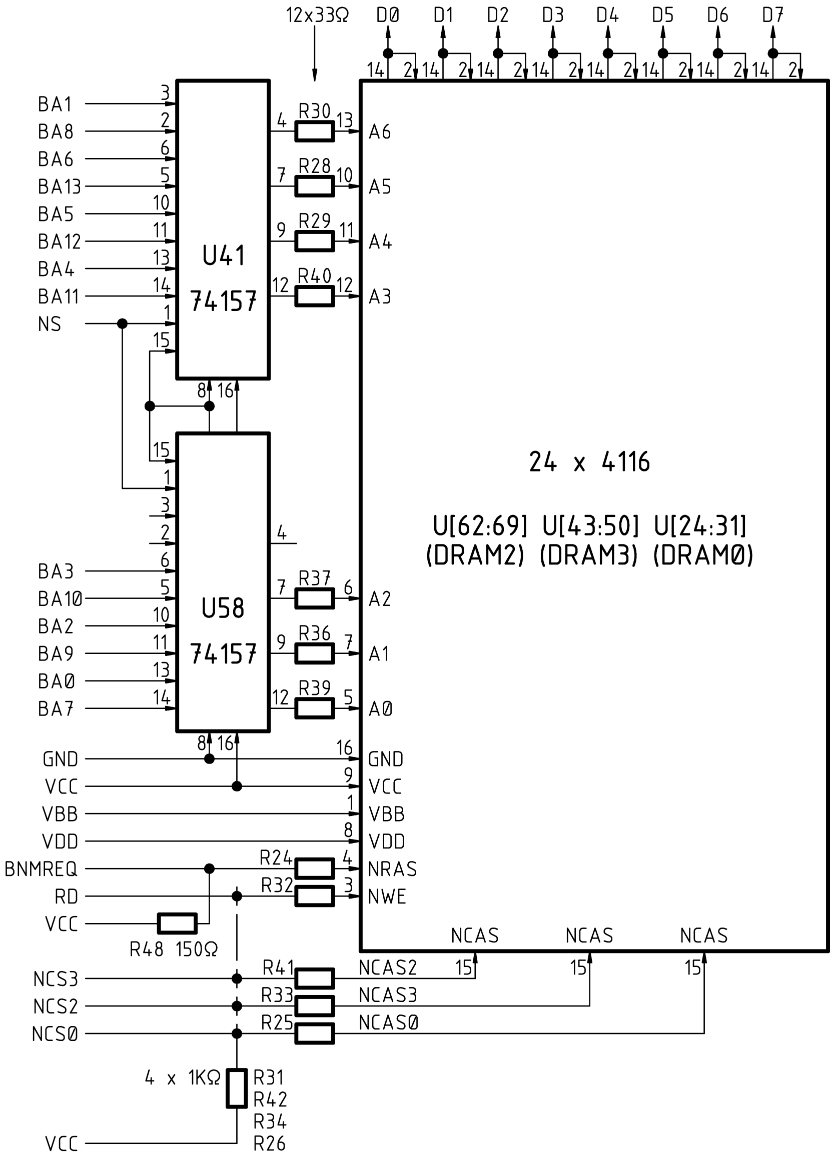


Fig. A1.4 – CoBra Microcomputer. DRAM Memory Circuit

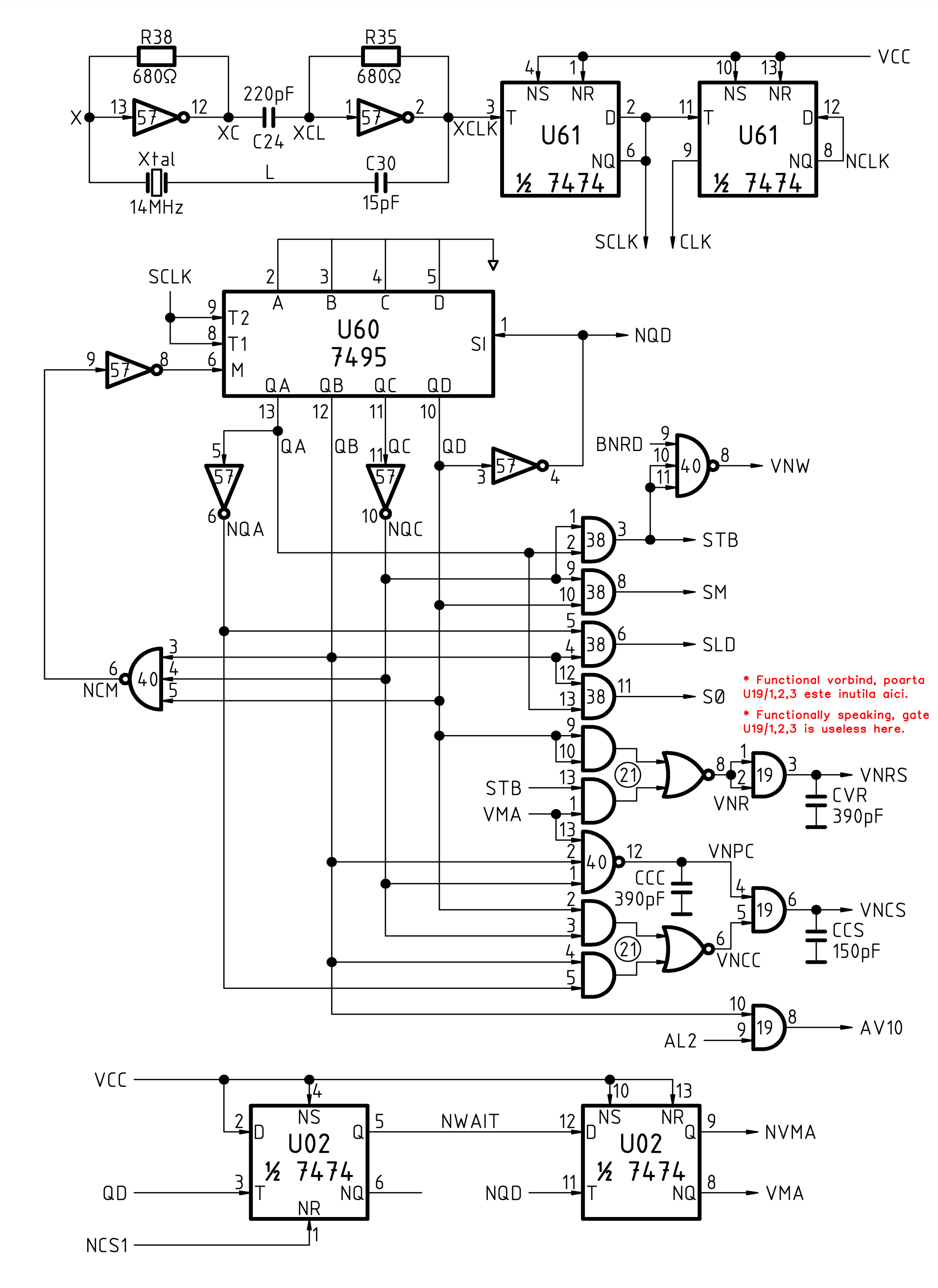


Fig. A1.5 – CoBra Microcomputer. Memory Access Prioritizer and Command Logic

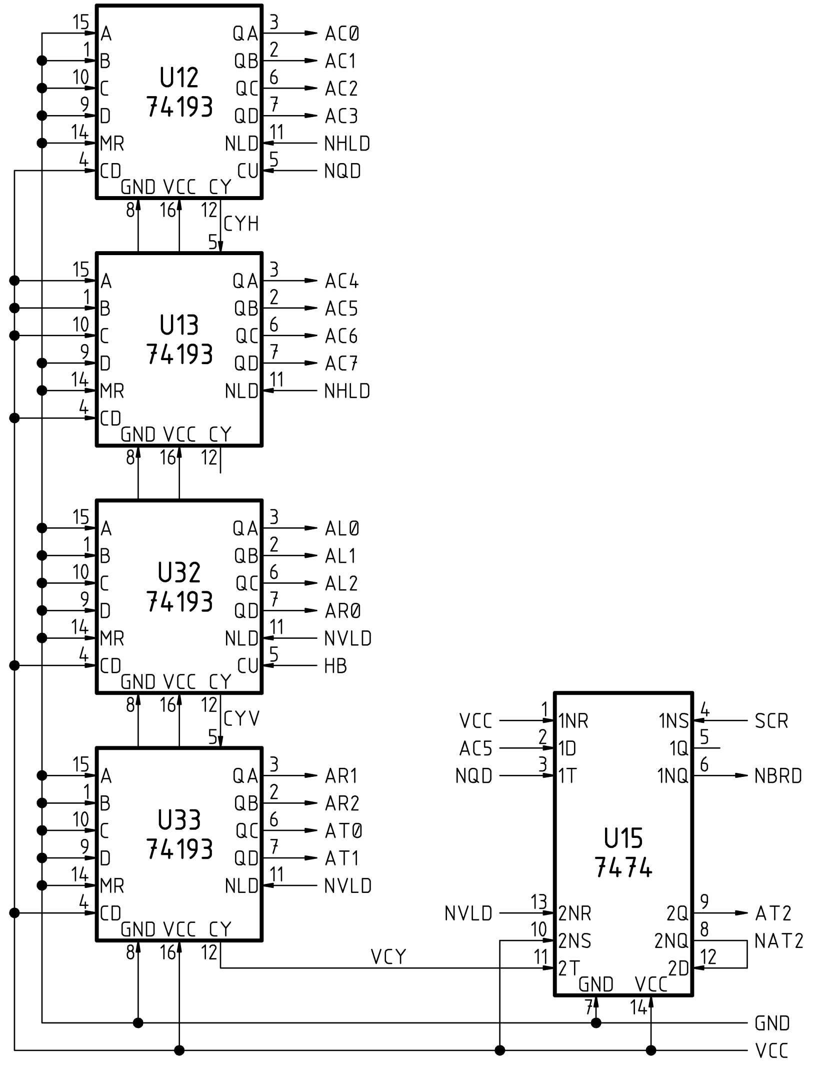


Fig. A1.6 – CoBra Microcomputer. Video Address Generator Circuit

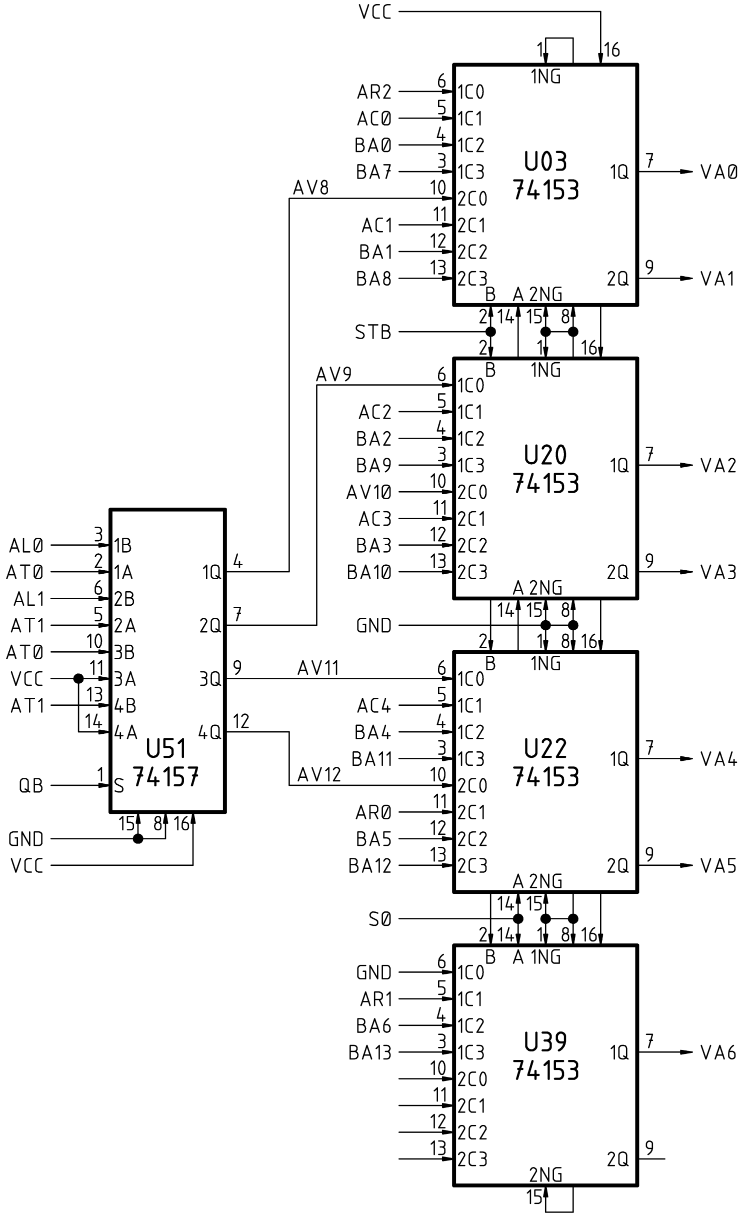


Fig. A1.7 – CoBra Microcomputer. Video Address Multiplexer Circuit

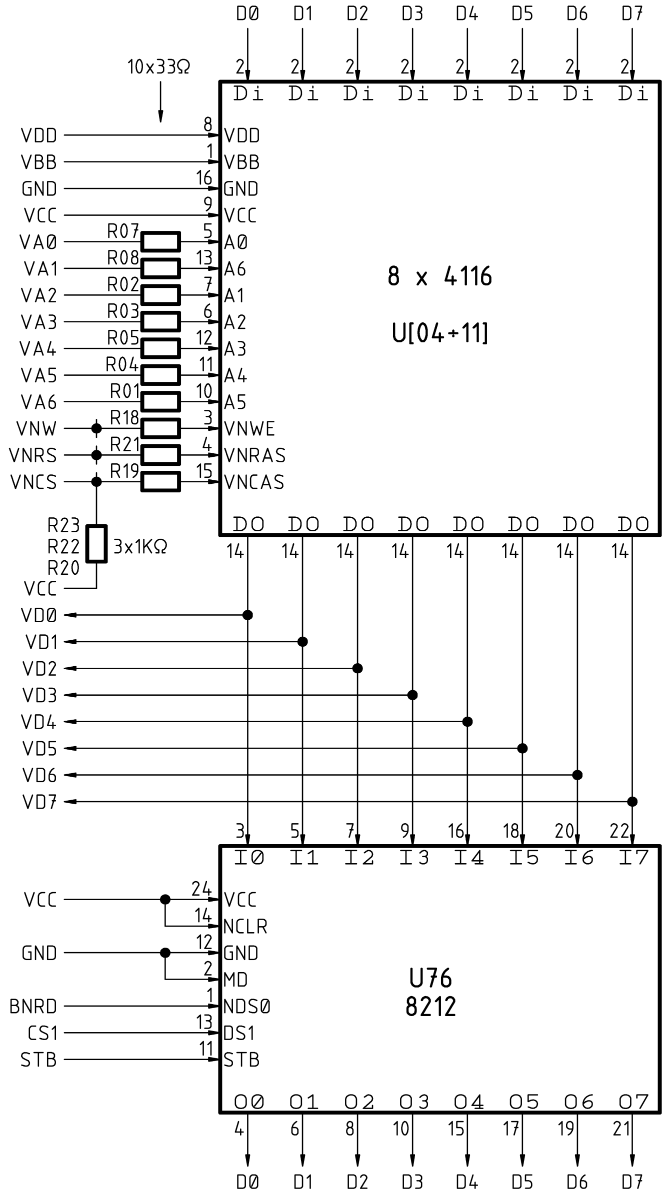


Fig. A1.8 – CoBra Microcomputer. Video Memory Circuit

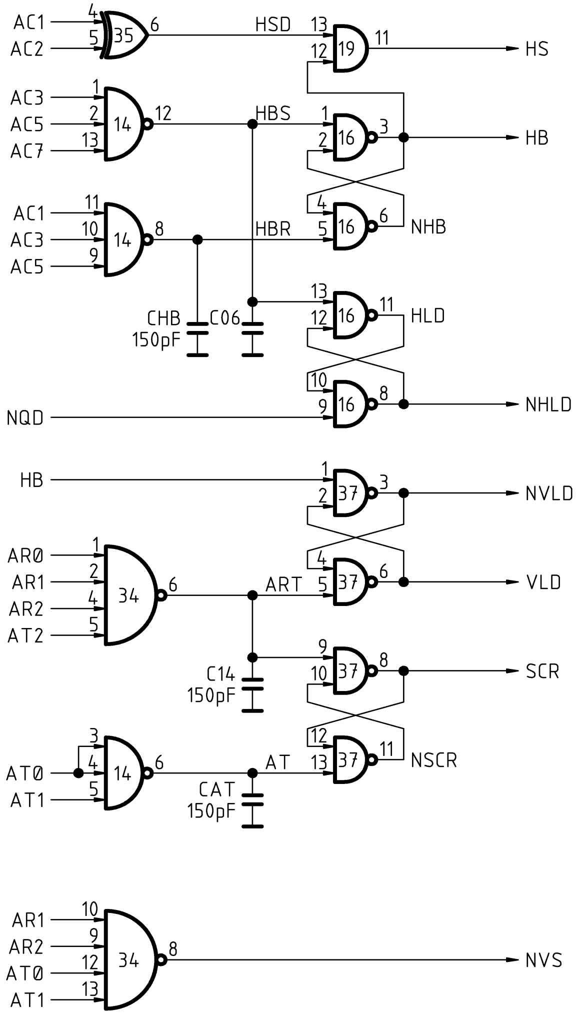


Fig. A1.9 – CoBra Microcomputer. Video Sync Pulses Generator Circuit

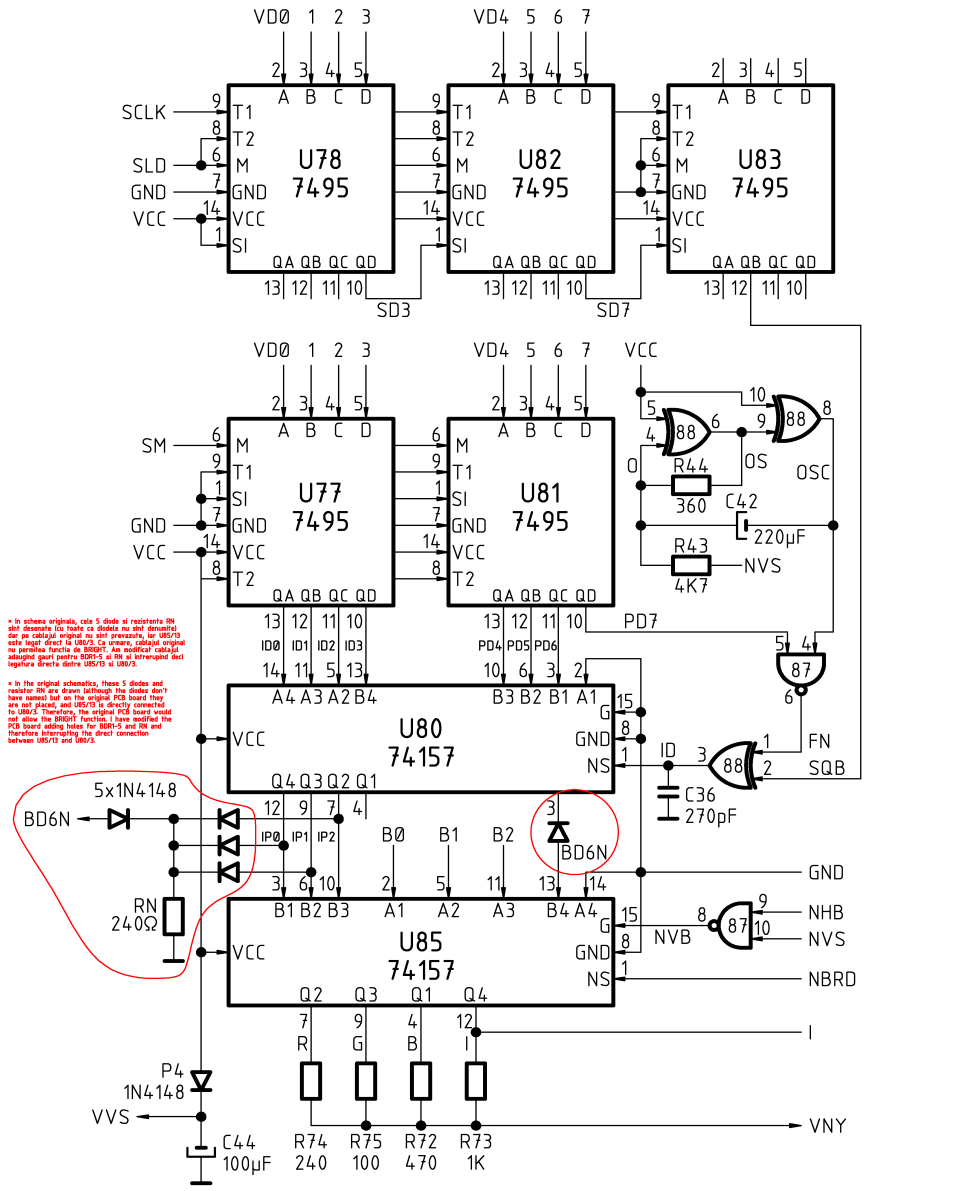


Fig. A1.10 – CoBra Microcomputer. Video Signal Shape Generator

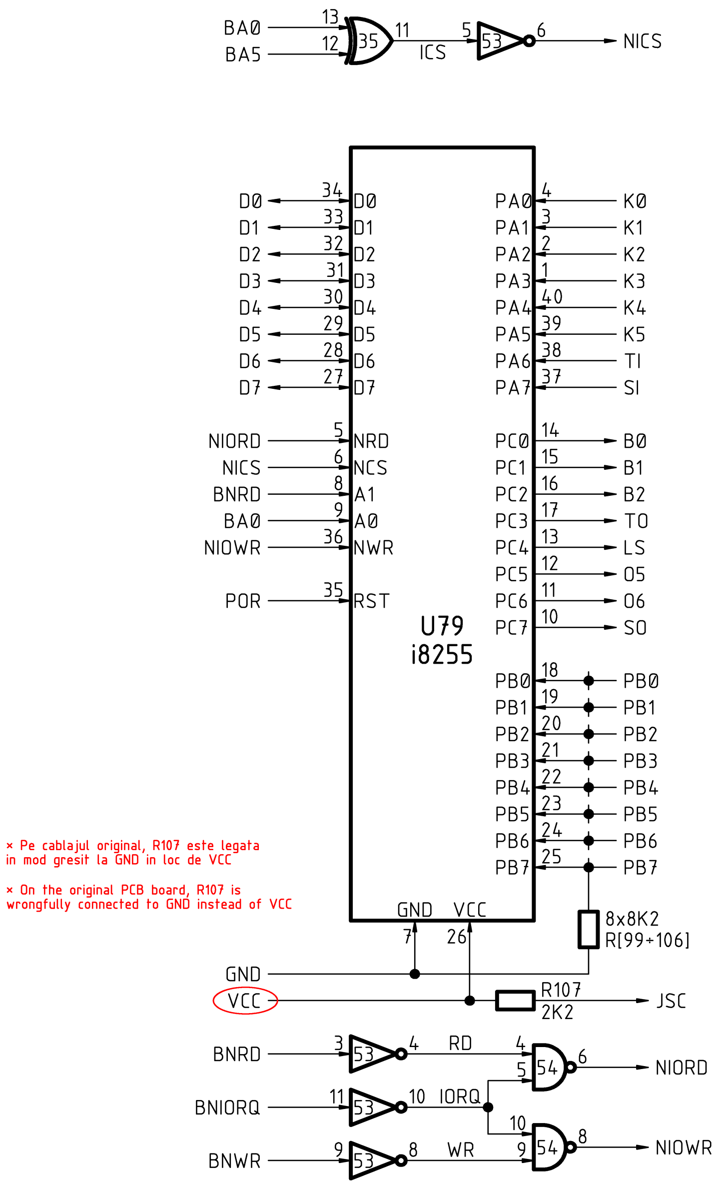


Fig. A1.11 – CoBra Microcomputer. Interfaces

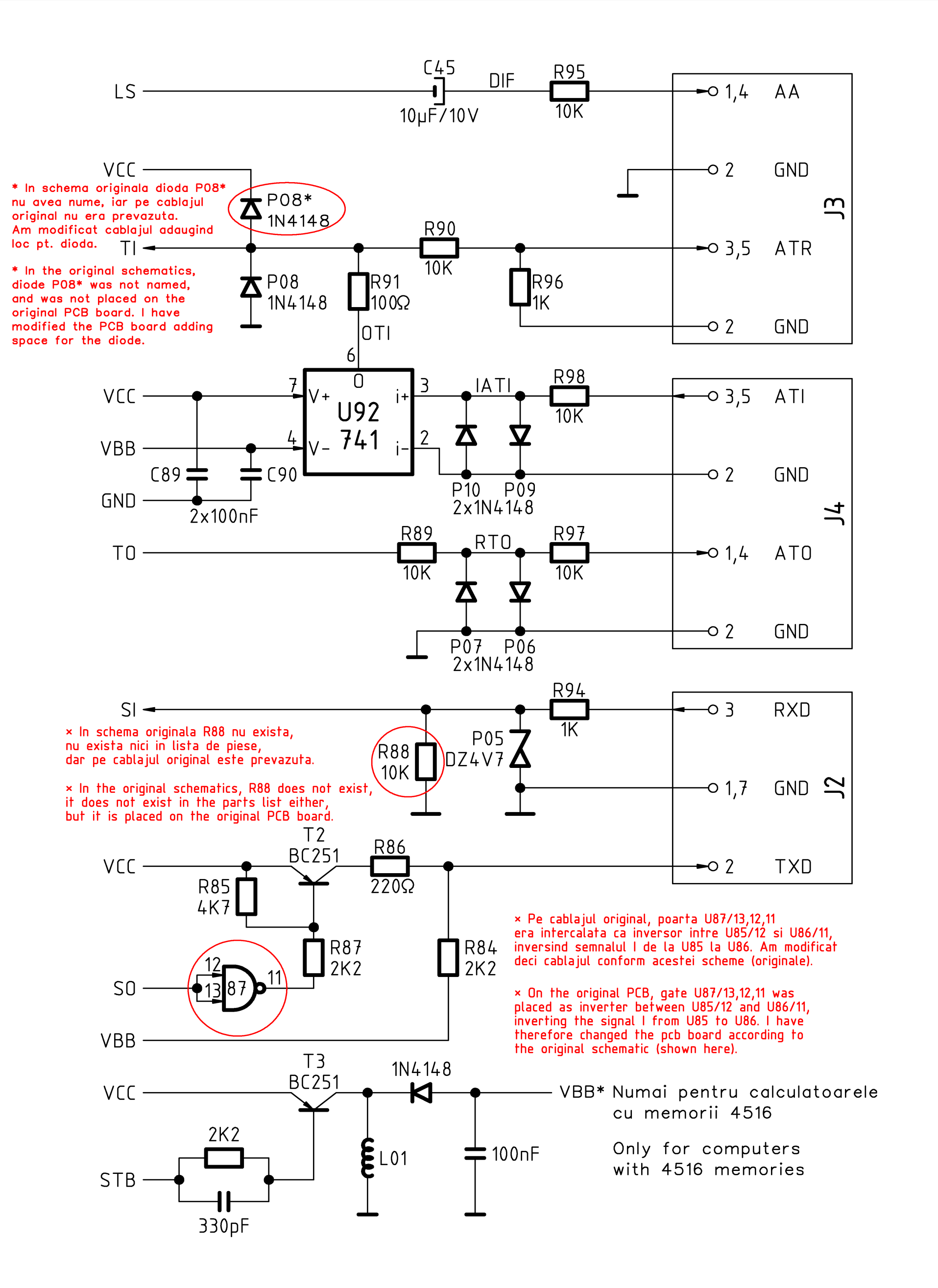


Fig. A1.12 – CoBra Microcomputer. Voltage-Level Adapter Circuits

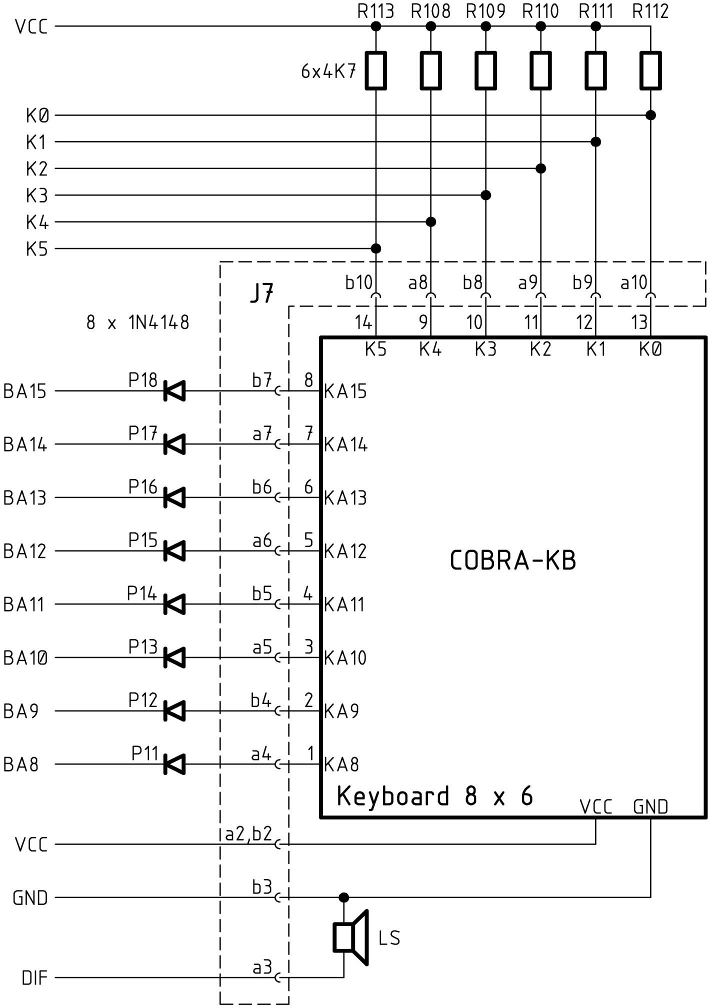


Fig. A1.13 – CoBra Microcomputer. Keyboard Interfacing Circuit

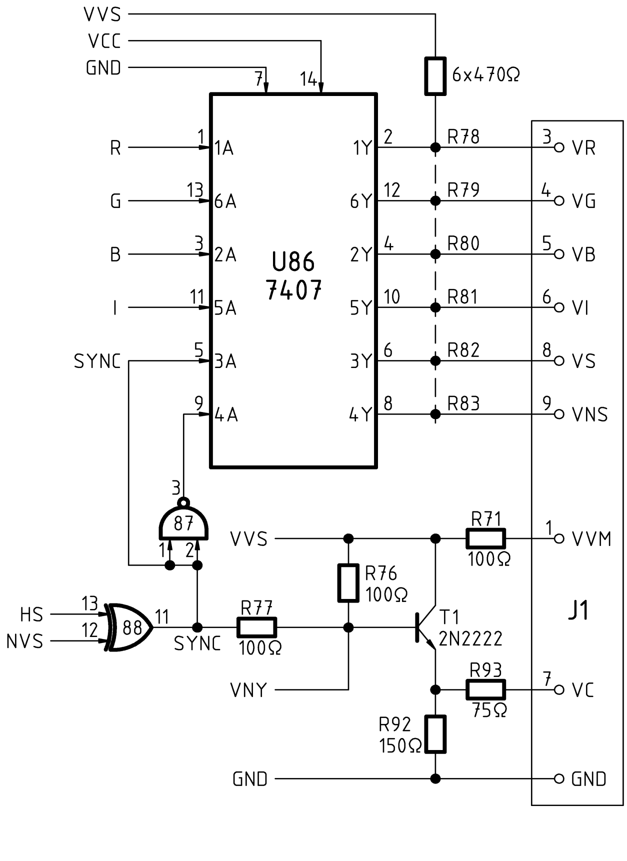


Fig. A1.14 – CoBra Microcomputer. TV Monitor Interfacing Circuit

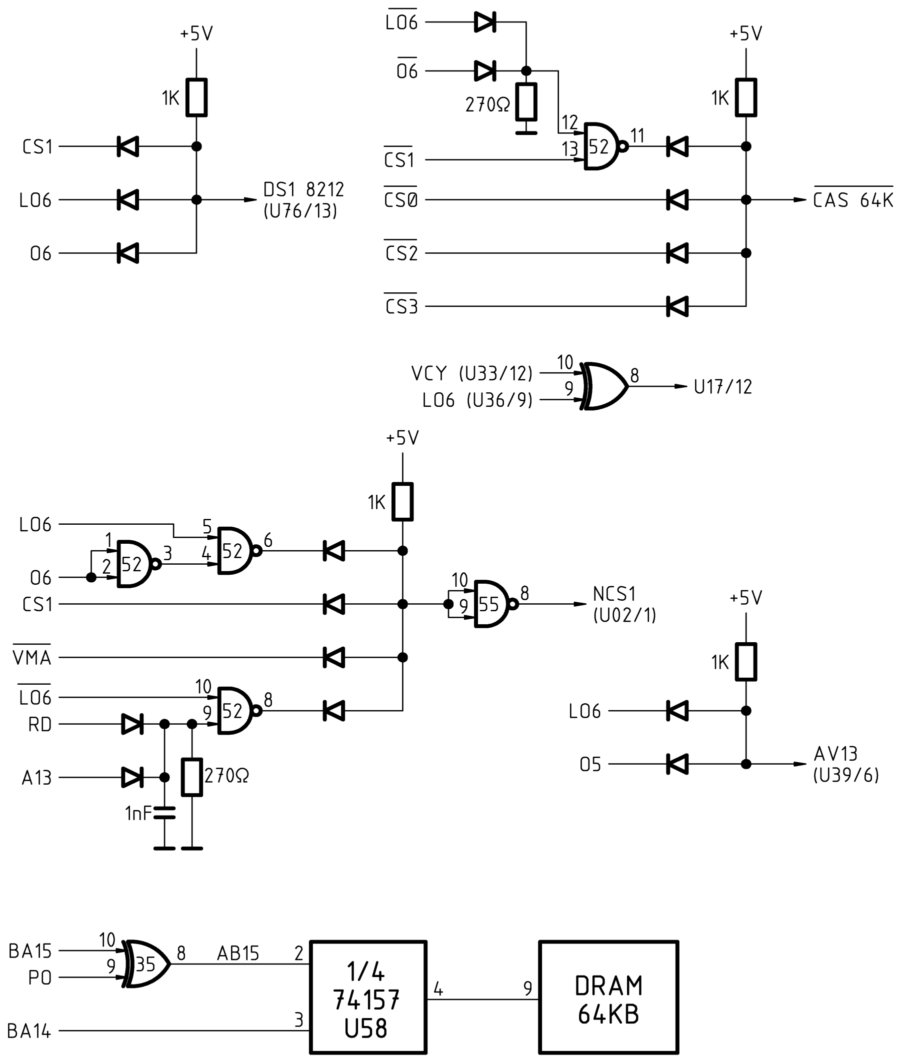


Fig. A1.15 – CoBra Microcomputer with Extended RAM Memory 16+64KB.
Changes to the Memory Configurator and Selector Circuit.

10. APPENDIX 2

Flopppy Disk Interface Schematics

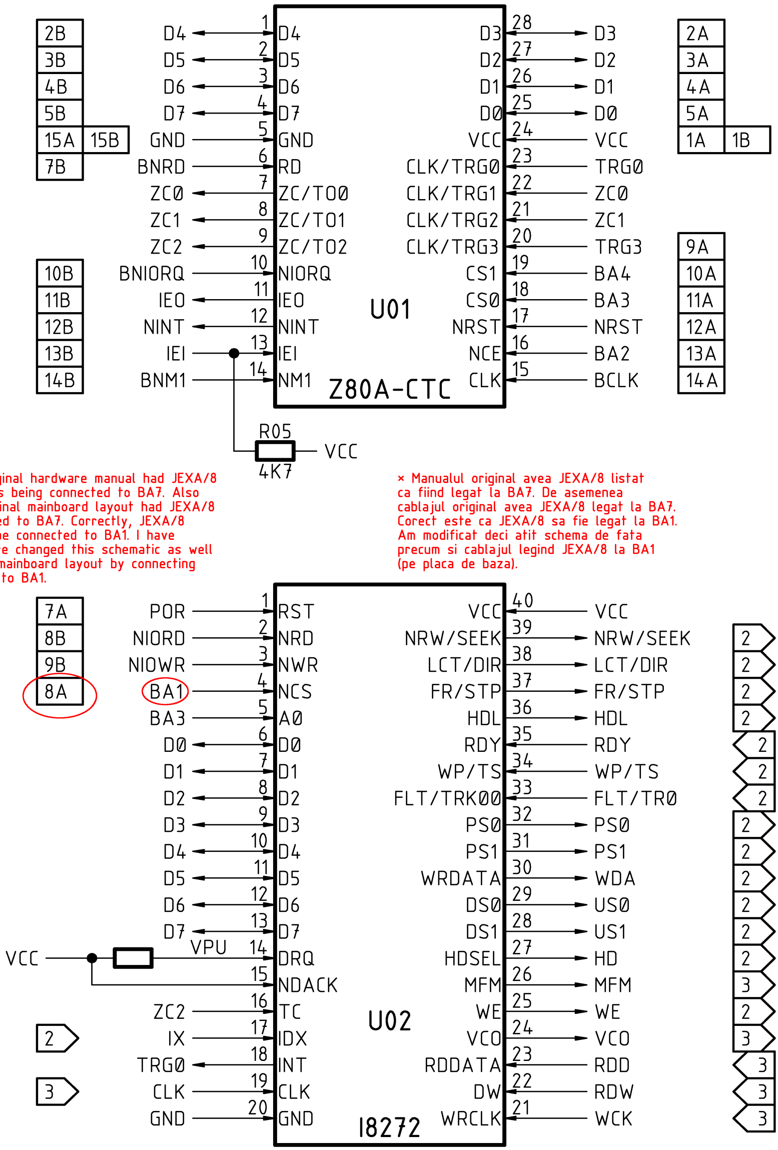


Fig. A2.1 – Floppy Disk Interface. Disk Controller.

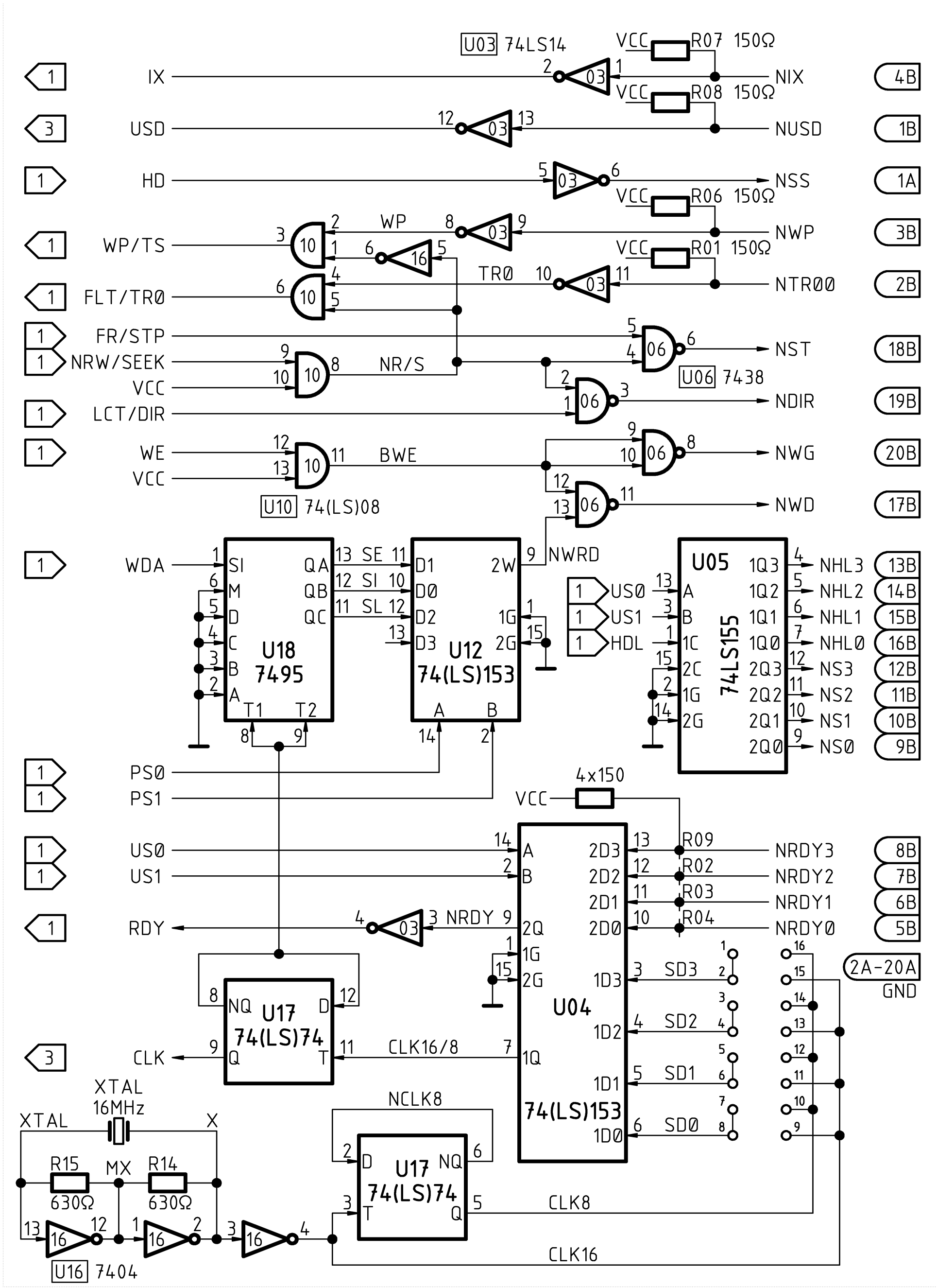


Fig. A2.2 – Floppy Disk Interface.
Command and Control Signals Generator Circuits.

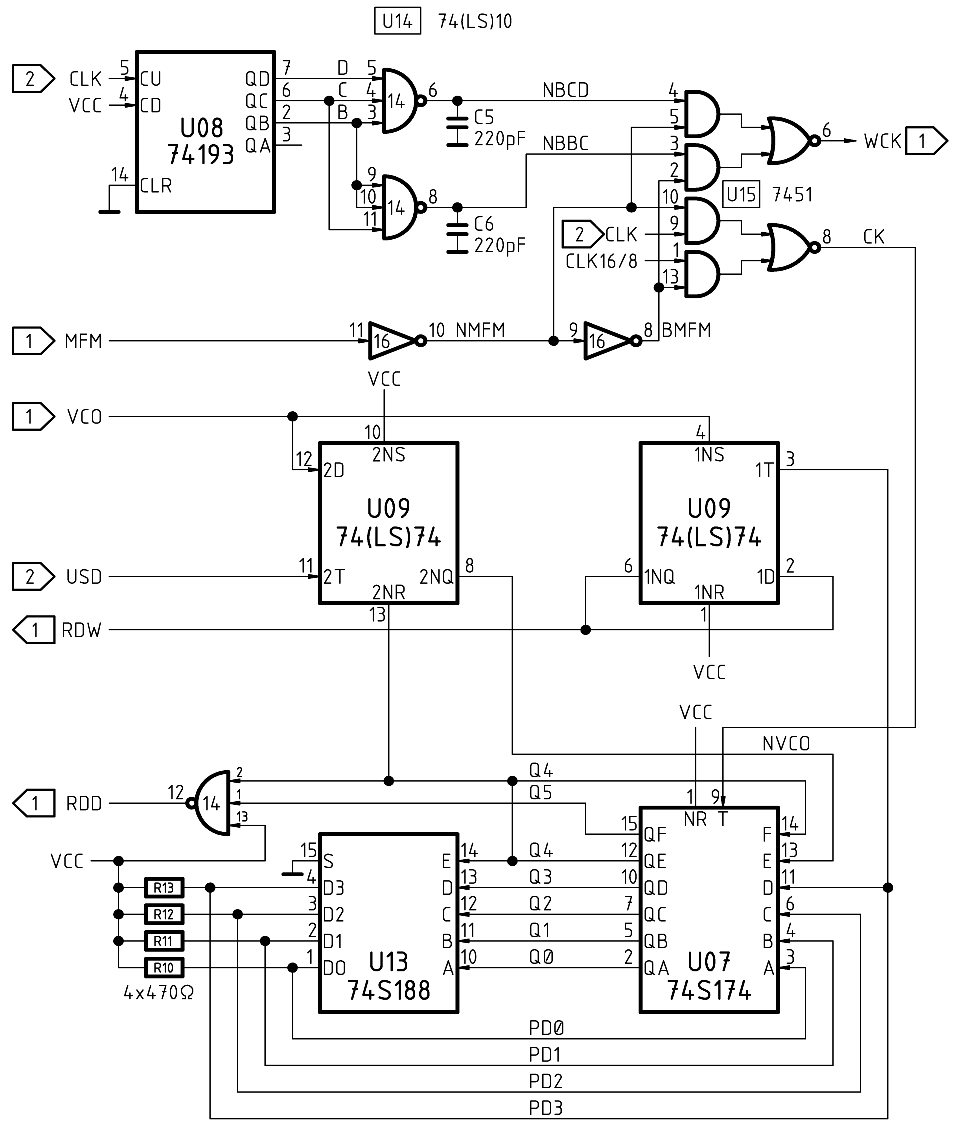


Fig. A2.3 – Floppy Disk Interface.
Write Clock and Digital PLL Circuits.

10. APPENDIX 3

Component Placement on Boards

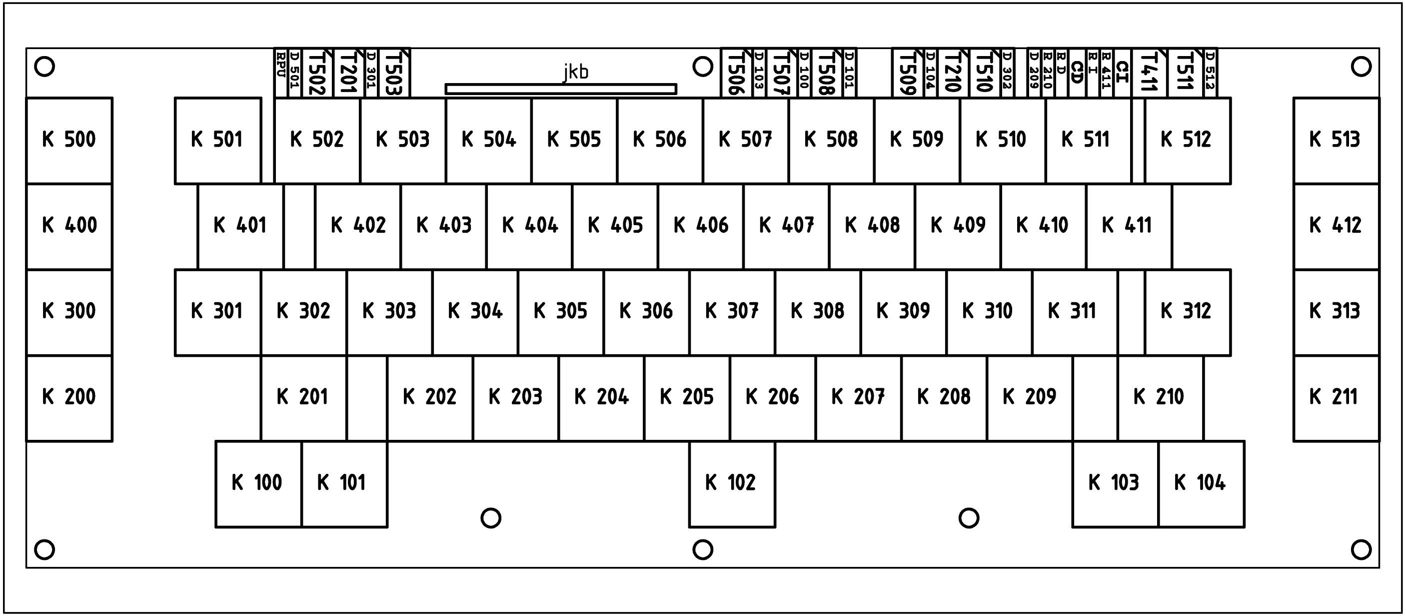


Fig. A3.1 - Component Placement on Keyboard Circuit Board

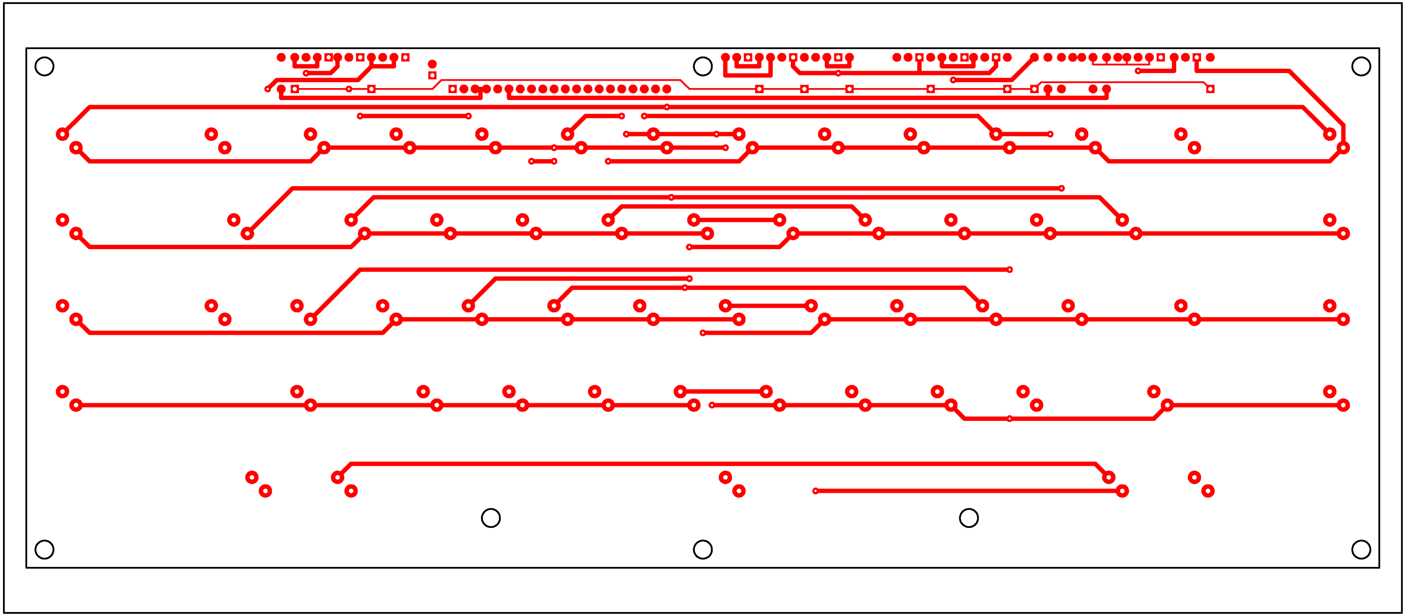


Fig. A3.3 - Keyboard Circuit Board - top layer seen from above (keys' side)

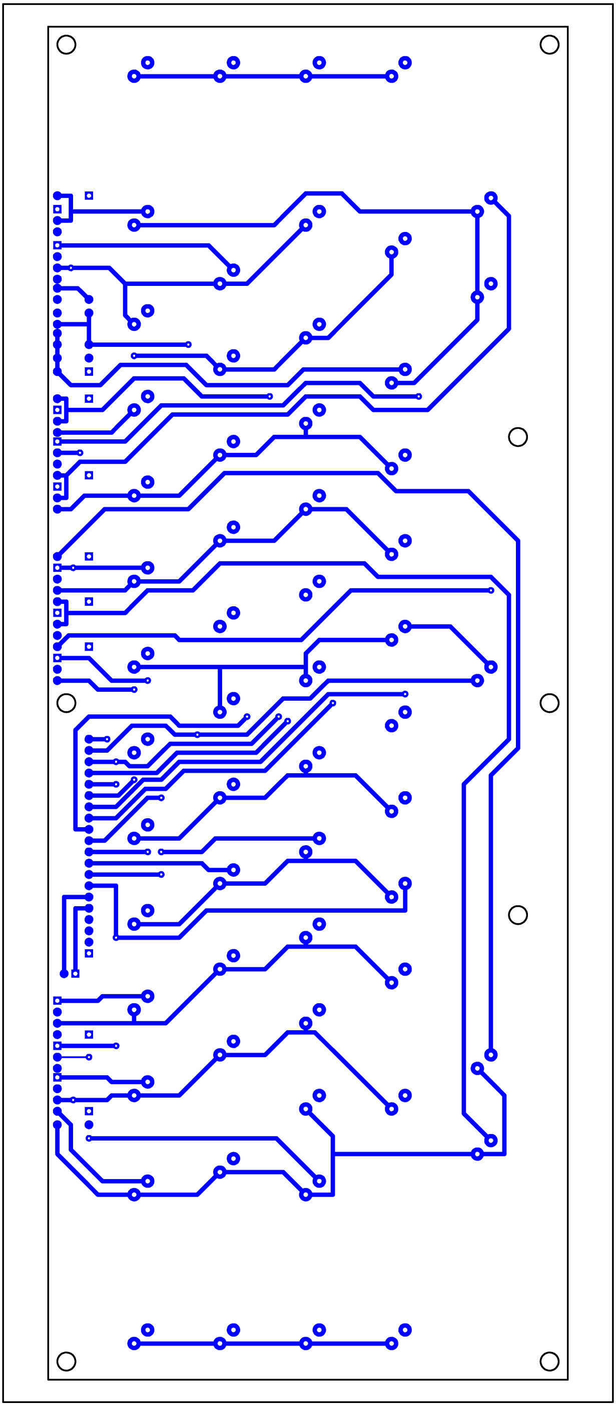


Fig. A3.3 - Keyboard Circuit Board - bottom layer seen from above (keys' side)

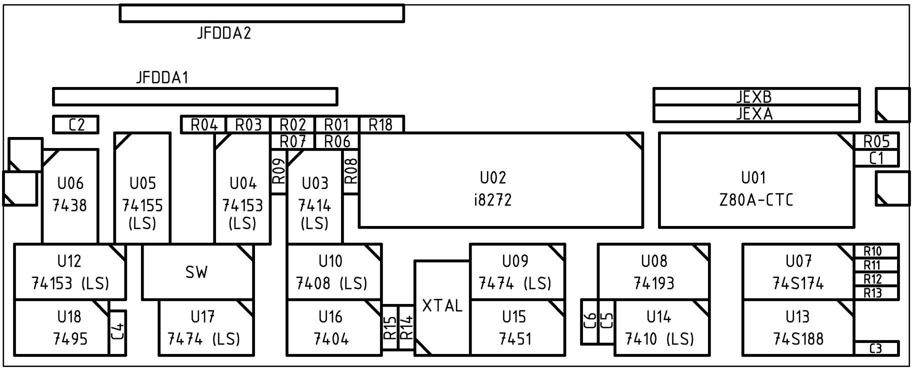


Fig. A3.5 – Component placement on the floppy interface board.